Daniel Song, Michael Lee, Aaron Liberatore
September 16, 2016

Project Design

Overall Goals and Feature Scope

- The general goal for the project is to provide a platform to automate communications between end-of-life patients/caregivers and their hospice nurses/doctors.
- Core concepts to focus on are the following: accessibility/ease of use, simplicity, portability:
  - Overall: Application is two-fold. We need to create 2 different apps for 2 different end users: caregivers and their respective hospice professionals. Smooth and quick UI/UX is essential.
  - Accessibility: Important to remove stress for caregivers; Important to speed up adoption time for hospice professionals
  - Simplicity: Important for caregivers who may be older individuals; Important for all users to ensure product's efficacy
  - Portability: Adds to product robustness, ability to use in diverse situations (ie. no wifi access).
- Strategic Goals: Create a robust MVP that can be marketable for DUKE CANCER INSTITUTE grants in future.
- Technical Goals: Fast and reliable data transfer
- Ethical Goals: HIPPA compliance
- Quality Goals: A usable prototype. Robust. No real quality goals currently besides having a usable MVP.
- Link to Full Product Design session in our last meeting 9/15

Design Goals:

Our solution for the Share team involves two distinct types of users, who have related but ultimately different experiences using the app. On the one hand, we have caregivers, who are using the app on behalf of their loved one, and on the other hand, we have the hospice nurse/hospital care team, who use the app to increase the effectiveness of their patient interactions. For each of these users, we have several features that we view as vital and core to the app's problem solving potential.

Core features for Caregivers:

- Patient status - Can use the app to answer questions about the condition of the patient, which are ultimately used to determine patient priority for the hospital team.
- Offline use and online sync - App can be used (e.g. answer questions) without an internet connection, and will update the server correctly once an internet connection has been established.
- History - Caregivers can see a history of all the answers to questions they have given on behalf of the patient.
- Communication - Some form of interaction with the hospital team will be possible within the application.

Core features for the Hospital Team:

- Patient priority - The hospital team user will be able to see a list of their patients sorted based on an algorithm to determine and display the patients most in need of their care and attention based on the caregivers' answers to questions about patient status.
- Customizable filters - The criteria by which patients are given priority (e.g. the deltas in their pain score over a short period of time) should be clear and transparent to the hospital team user, and should also be customizable.
- Daily routing and schedule display - The hospital team will be able to see all of their appointments with patients for the day.
- Communication - Some form of interaction with the caregivers of each of their patients will be possible within the application.
- Exporting of information - Hospital team users should be able to export useful information in the app so that it can be used elsewhere (e.g. export the patient priority list).

What will be easily changeable:

- Changes to names/text fields, and small changes to wording (but not the overall content).
- Changes to font sizes and styles, and changes to colors/appearance.

What will require programming expertise:

- Any additional features or screens, especially if it needs to continue allowing for offline use but has an online component.
- The set of questions that caregivers may answer on behalf of their loved one. This requires updates to not only the user interface but also updates to the prioritizing algorithm and to the types of customizable filters.
- Any fundamental changes to existing features or their implementation. For example, if we opt to only allow caregivers to communicate via twitter-length messages, there will need to be code changes if you suddenly wanted to change it to a instant-messenger experience.

What the client will need to be responsible for in the future

- If the user base grows large, Firebase (the backend service for the app) will no longer be free, and the client will need to evaluate if they want to pay the monthly cost.
- Because React Native allows for the development of both Android and iOS apps, the client may need to take steps to ensure the app remains functional for both of these mobile operating systems as updates or major version releases are pushed out.

Impact:

Because of the rising costs of healthcare and the prevalence of insurance plans that only cover one or two short hospice visits a week, the need for an easy and effective communication system during hospice is becoming increasingly more important. The current process for hospice care revolves around an out-dated and fragmented communication line between caregivers and their hospice team. Centered around a third-party liaison responsible for triaging patient information to hospice nurses, this disjunct and infrequent communication channel often results in unneeded stress caused by conflicting, incomplete, and lost information. Rather than have a disconnected third-party collect and transmit information from

multiple sources, Share aims to streamline this communication channel giving caregivers the access to the hospice team they want and the hospice team the specific and up-to-date information they need.

From the perspective of the caregiver, Share will alleviate stress by giving them the comfort of knowing their doctor is always up-to-date on their patient and providing a means for direct communication with their hospice team in order to get the most appropriate care during their visits. Yet, through this streamlined process, the hospice team may receive even greater benefit. Instead of receiving conflicting, redundant, or extraneous information that itself has to be deciphered, the hospice team will be able to instantly monitor all of their patients having the most critical, curated information at their fingertips. Through aggregating statistically significant data points, Share will be able to automatically prioritize patients allowing nurses to better schedule visits and more effectively treat patients during them. By incorporating Share into the hospice care program, caregivers and nurses will reduce stress levels and more efficiently care for their patient.

Dependencies:

The application itself will be dependent on the React Native development framework and an instance of the Firebase Cloud Service. In order to build the content of the application, the team expects our client to determine the question set and flow that will be used to calculate a patient's daily health status. The client is also expected to supply a general algorithm or statistical method in order to determine when a patient is in a critical state. Along with the content, the client is expected to help connect the team with test users; either past hospice caregivers or people closely connected with hospice home care.

Concerns:

- User interactions:
    - Due to HIPPA patient privacy laws, it will be difficult to interface with the actual end users. The best that we can do to do user testing would be to approach former caregivers. We might also have to plan far ahead to interface with hospice professionals who run a tight schedule
- Cost:
    - This is not a large concern. The MVP will serve at most 1000 patients on a long time horizon. Database costs are not huge for such a user base. Other costs are small in comparison (Apple Developer licensing)

Team Organization:

- Product Manager: Michae
    - Taking charge of product development and communications with the Duke Cancer Institute client.
- Teach Lead: Aaron: Development Environment
    - Looking into React Native environment.
- QA: Daniel look into app testing environment
    - Looking into application testing suites