Spring 10

Observations of Daily Living

Abrahm Coffman, Matt Gedigian, Annette Greiner, James Tucker, Nat Wharton

This project represents the design and development of an iPhone application, website, and supporting framework that simplifies the collection and utilization of observations of daily living by patients, parents, and physicians. These observations of daily living provide a wealth of information for patients to use as preventative measures and in managing chronic conditions. Physicians can use that same information to refine diagnosis and treatment. The KidOoDL website and iPhone application, designed around primary stakeholder needs –parents with infants and pediatricians- take advantage of appropriate technologies, providing innovative yet intuitive tools for tapping a hidden medical resource and bettering patient health outcomes.

Contents

| PROBLEM STATEMENT | |
|---|----|
| PROJECT GOAL | 5 |
| USER REQUIREMENTS | 6 |
| NEEDS ANALYSIS | 6 |
| COMPETITIVE ANALYSIS | 6 |
| USER INTERFACE DESIGN | 8 |
| METHODS | |
| DEVELOPMENT OF IPHONE DESIGNS | 9 |
| DEVELOPMENT OF WEB DESIGNS | |
| FINAL IPHONE DESIGN | |
| FINAL WEB DESIGN | |
| VISUALIZATIONS | |
| MILESTONES | |
| OBSERVATIONS OF DAILY LIVING | 20 |
| System Design | 23 |
| PLATFORM ARCHITECTURE | 23 |
| DATA ARCHITECTURE | |
| SCALABLE ARCHITECTURE | 25 |
| ADDENDUM | 26 |
| THE BUSINESS MODEL | 26 |
| PATIENT VALUE PROPOSITION | |
| PROVIDER VALUE PROPOSITION | |
| Pricing & Growth Strategy | 27 |
| APPENDIX I | 28 |
| APPLICATION FRAMEWORK DECISION ANALYSIS | |
| SWOT ANALYSIS I | |
| SWOT ANALYSIS II | |
| SWOT ANALYSIS III | |
| WEB FRAMEWORK DECISION: PYLONS | |

Problem Statement

Observations of daily living are "patterns and realities of daily life" including but not limited to "diet, physical activity, quality and quantity of sleep, pain episodes and mood.¹" However, collecting, understanding, and using this information is time consuming and cumbersome. By applying modern Internet and communication technologies to patient centered healthcare, this project helps patients and caregivers collect, share, understand, and best utilize their information.

During a clinical visit, physicians have a limited opportunity to assess a patient's condition. In addition to their own examination, they can ask questions about the patient's experience since their last visit. Key determinants of health take place between these episodic encounters and a physician's ability to act on them is dependent on this information being communicated effectively. As patients, we do our best to remember, but being able to properly recall every instance of our cause for going to the physician, especially when a chronic condition, is incredibly difficult to do over long periods of time.

Many people benefit from collecting ODLs. Chronic disease patients, pregnant women, parents with new infants and children are just some for whom ODL collection can provide actionable information for bettering their health outcomes. Patients with diabetes already know the value of tracking their body's response to food consumption by time, type, and amount and measuring that against exercise. This allows the patient to better decide insulin dosage in a way that requires less physician intervention. People living with HIV follow similar tracking regimens to keep their bodies healthy in the hopes of avoiding toxic drugs until absolutely required. Tracking ODLs helps the patient feel more in control of their condition and life, possibly leading to better outcomes. Tracking the information motivates the patient to act proactively rather than just wait for a physician's diagnosis and treatment. It is important to note that not all ODL tracking should be done after a problem occurs. For example, parents of newborns track information as a preventative measure, not after the baby becomes ill.

Observations of daily living are also useful to physicians. While the validity of measurements taken by patients at home may be open to question, if there is a consistent pattern presented to the physician in a clear manner, he or she has that much more useful information, qualitatively and quantitatively to make a diagnosis and design a treatment plan. When this information is tracked more systematically and perhaps at the behest or following the advice of the physician, the physician is much more likely to incorporate patient created information in the diagnosis and treatment plans. The information also provides the physician a more holistic or 360-degree view of the patient's condition and possible unforeseen causes or triggers.

Tracking observations of daily living is often difficult, cumbersome, and time consuming. Only the most dedicated are likely to carry around a notebook all day, hand writing everything one wants to track. Condition-specific devices can help, but also add to the clutter of devices one carries each day and can signal that person's illness to others in ways the patient may want to avoid or resent. In addition to the physical problems of recording this data, there are reporting and recall biases that affect efficacy. When one does not record the information as it occurs, it is very common that the information is wrong, even

¹ Robert Wood Johnson Foundation. "Feature: The Power and Potential of Personal Health Records." January 2010. Web. 2 May. 2010. http://www.rwjf.org/pr/product.jsp?id=51248

if just incrementally. These small increments have a large impact on aggregate data validity. One may also need to take the collected information to a physician or caregiver to properly understand its meaning.

With this in mind our project has focused on addressing the need for ease of collecting, storing, sharing, and analyzing observations of daily living. Our use case, tracking information about infants, also includes tracking developmental milestones.

The target audience for our project is any person who can benefit from tracking ODLs, patients, family members, caregivers, and physicians. As a use case we've focused on assisting parents with newborns in tracking important developmental information about their babies. This is especially salient to one of the team members who recently brought home a five-week premature infant for whom ODL and milestone tracking was of paramount importance.

When parents leave the hospital with their newborn, their physician commonly assigns them a set of ODLs to track about their child. From the moment the baby is brought home, parents need to track how much and when the baby eats, both bottle and breast feeding, how much and how often the baby sleeps, and frequency and type of diapers (urine, feces). This information, provided to the physician, details the baby's development. If the child falls too far outside the norm there may be a problem and the parents must quickly respond by contacting their physician. Tracking this kind of information and having an average or norm to expect offers a great deal of comfort to first-time parents and parents of premature infants. It also acts as a signal to action if there may be a problem. Knowing their baby fits within a normal range goes a very long way towards helping them manage their new, much busier lives.

In addition to collecting eating, sleeping, and elimination ODLs, developmental milestones are key indicators to physicians especially about the child's growth progression as it matures. There are specific abilities babies should develop within a range of time (i.e. within the first three months an infant should be able raise its head and chest while lying prone). If the child does not achieve these milestones, this may signal a problem to the physician.

Our solution to the challenge of collecting, managing, sharing, and maintaining ODLs and milestones is a soup to nuts service that includes an iPhone application, website, web service that synchronizes the mobile application with other mobile applications as well as the website. Our service will allow multiple caregivers to input and view information about the child, provide meaningful visualizations for the parents and the child's physicians, as well as link to personal health record (PHR) systems such as Google Health. By supporting the storage of data in Google Health, our service not only provides greater control and continuity to the patient over their medical information, it eases integration into electronic medical records (EMR) on the physician/hospital side.

By combining the collection of ODLs and milestones our service provides a huge opportunity for individual users to track patterns in their health. By collecting this information anonymously across a large swath of people, we may assist in achieving greater insight into child development overall.

Project Goal

This project had three interconnected goals.

The first was to design and develop a working iPhone application that meets the needs of parents and caregivers in collecting, viewing, sharing, and understanding their baby's ODL information and milestones.

The second goal was to create a web application that works in conjunction with the mobile application, syncing information from the web application to one or more mobile application. The website would allow multiple caregivers controlled access. The website and mobile application would both provide an opportunity for parents to find patterns. An example of this is finding a correlation between eating and sleeping habits via data visualization. The website would also provide parents decision support and physicians meaningful measures of development quickly, easily, and intuitively. Finally, the website would allow users to write their data to third party PHR vendors such as Google Health.

Our third goal was to develop an overall framework and API (application programming interface) for the collection of multiple types of ODLs and milestone-like information. A scalable and flexible data model is imperative in achieving this.

Our use case for this project was the KidOoDL (Kid Observations of Daily Living) iPhone application and website. While this application stands well on its own, we plan to expand the types of ODLs collected and have included this intent at every step of design on the framework, data model, and APIs. We built our system from the ground up to allow scaling. We hope to convince others to use our framework and API to create applications that connect to our service for other conditions and needs. We built the framework with the intention of encouraging voluntary social exchange of information by users.

Each team member had individual goals in contributing to this project. The overarching goal for all members was taking the opportunity to utilize and practice the skills we developed during our two years at the UC Berkeley School of Information in a real-world setting. We wanted to create something tangible that would last beyond fulfilling our degree requirements. We each were drawn to improving healthcare and found the nexus of our skillsets created an opportunity to use our newly gained knowledge to improve healthcare outcomes in a way that is cutting edge, yet feasible.

User Requirements

Needs analysis

In order to gain insight into the needs of our potential users, we conducted face-to-face interviews with new parents; five couples with infant children. Two other couples provided written responses to interview questions. Additionally, one member of our project team is a parent of a premature infant. To understand the needs of pediatricians, we met with a community pediatrician, two pediatric hospitalists, a pediatrics resident, and a focus group of academic pediatricians. The interviews were supplemented by two surveys, one for parents and one for physicians (n=10 for each survey).

In studying new parents, we learned there was a fair amount of variability in parenting styles and thus in needs for an application to support the tasks of parenting. Parents varied in how interested they were in formally tracking data for their baby, which things they chose to track, and which observations they made about the things they tracked. They all agreed on the importance of doing some tracking, many having been encouraged to do so by their pediatrician. The parents tried a variety of techniques for tracking, most with paper or some type of journal. Parents who used paper-based tracking felt that it was difficult to manipulate (particularly at night) and easy to misplace. Several had tried electronic applications, either web-based or smartphone-based, but none were satisfied with their chosen tools. The prime reasons for discontinuing use of electronic applications were poor usability and lack of necessary features.

Parents were universal in their need for an application that could be used when half asleep, in the middle of the night, with minimal fumbling. They also drew attention to the need to allow multiple caregivers to enter data. We found the most universally tracked observation types were feeding (both breast and bottle feeding), diapering, and sleeping. Many parents also wanted to be able to enter comments associated with their observations. Most parents had smartphones, and 60% had iPhones.

We found that pediatricians, like parents, varied in their interest in tracking baby data. Some felt that it only needs to be encouraged for infants with known problems and that parents would be better off not worrying about tracking. Others felt that all new parents should track a specific subset of data. In addition to eating, sleeping, and diapering, pediatricians were very interested in adding milestones to our list of observation types. The connection between parents and pediatricians is central to our product concept. Therefore, we feel including milestones will make our system more helpful to the medical community, and by extension, parents and infants.

Competitive analysis

There are several existing smartphone- or web-based applications for tracking baby data. We looked at Baby Brain, BabyConnect, iTrackBaby, Baby Activity Logger, NursingLog, Total Baby, Bant, Weightbot, and Baby Geek for the iPhone, Baby ESP for Android, and the web-only Trixie Tracker. These applications vary in the subset of observations they include, from nursing alone to a list of thirteen categories of data. Most accept data for two to four types. Only two applications accept data via a web site. Both of which have HTML-based applications for the iPhone as well. One offers web pages optimized for view on a mobile browser, while the other has an iPhone application built with HTML. We also looked at web sites that collect other types of observations of daily living from users. One such site is Keas, which gives users modular "care plans" tailored for those with a specific illness or interest. Keas care plans are developed with physicians, giving the site a certain level of authority. Users can subscribe to the care plans of their choosing, a model of customization that we find practical.

Another ODL collection web site, theCarrot.com, allows users to track a wide range of things about their daily life, such as body measurements, foods, energy level, and exercise. The site provides customized reports and options for sharing data with a community of users.

PatientsLikeMe allows patients to share information about their experiences with having a specific disease. They input information about their symptoms and medications, sharing that data with other users or even the general public. This site also reaches out to the medical research community making data available for studies and helping patients find studies to join. We would like to incorporate similar social networking features in a future version of our product.

We have also been following the development of the Robert Wood Johnson Foundation's Project Health Design. This program has granted funds for research in collection of observations of daily living and integration with personal health records. In particular, the FitBaby project is relevant to the KidOoDL concept. The team is developing a mobile device for parents with low-birth-weight, pre-term babies. It aims to deliver near-real-time data to clinicians, including data about the infant as well as the parents.

User Interface Design

Methods

Our research among parents made it clear at the outset our application's usability would be central to adoption. We chose to follow a user-centered design process to ensure we built our application and service to meet user needs in both UI design and features. The requirements called for a design for our iPhone application and complementary web site. The two shared most features, but the designs needed to be developed, in tandem, for each platform.

As discussed above, we began with interviews and surveys of parents and physicians. From this input, we created a set of personas: a father, Rick; his wife, Angela; a community pediatrician, Emma; an academic pediatrician, Scott; and a babysitter, Susan. We chose Angela as our primary persona, since her profile as a mother who is not working made her most likely to be caring for and tracking data for the infant (Sammy).

Next, we conducted a task analysis for our set of personas to determine which features were of highest priority. From the prioritized feature set, we developed scenarios in which the personas completed the tasks to be supported by the interface. Working with scenarios, we developed initial designs for both the iPhone application and the web site.

The designs were first produced in Balsamiq Mockups. We then used prints of the electronic files to generate paper prototypes. We conducted usability tests with the paper prototypes, iterating the design between testing sessions. In order to test the iPhone prototypes more accurately, we built a test device to hold a paper prototype screen that could be "scrolled" up and down with a thumb. (See figure 1.) We tested each interface three times. The first tests of the iPhone and web applications were conducted with different users, to avoid transference of knowledge from one test to the other. For subsequent tests, we asked the same users to perform tasks with both interfaces. This assessed how well the two worked together. We modified the paper prototypes after each testing session.



Figure 1. Paper prototype for iPhone application, with test device.

After conducting usability tests, we began graphic design work. First, we created a logo for the KidOoDL service, then detailed designs in Adobe Illustrator for the home page and the data entry page. We wanted to convey a sense of simplicity and ease while also appearing formal enough to be trustworthy. The color palette is a calming blue with accents of more lively green and red. The logo font, Reliq, is playful and active. The symbology is of growth and nature (grass, blue sky).

Development of iPhone Designs

For the first version of the iPhone design, we wanted to experiment with accepting data input in a single click once the application started. We chose to make the data entry screen (Figure 2) the root of the iPhone application. From that screen, the user could immediately save a single observation entry by tapping the "Add" button for that observation type, thereby accepting the defaults (current date and time, most likely option settings). If the user wanted to enter non-default data, each observation type had controls for setting common options on the top-level screen. Each observation type had links to a separate screen for setting less common options. In usability testing, we found that the single-click method of entering data was confusing for users. They were unsure when and whether their data had been saved. After testing several single-click options, we decided to use separate screens for each data entry type.



Figure 2. Initial iPhone data entry screens. The left screen appears as the application first opens. Center shows the sleep timer three seconds after the user presses the "Start" button. Right is the diaper data entry screen.

Each data entry screen also linked to screens for viewing that category's history, reports, and making configuration settings. Once the user navigated away from this screen, their movement through the application was restricted to returning to the root level. Usability testing showed that this was too restrictive. We therefore added tab bar navigation on all non-modal pages of the application.

The initial version of the design did not adequately handle switching from one child to another. Users were sometimes confused about which child's information they were editing. To address this we included identifying information on each data entry screen.

For back-end development purposes, we use the concept of a "chart" to identify the data object containing all observations for a single child. The first version of the iPhone design also identified children's data in the user interface as belonging to a "chart". Users found this terminology too medical and unrelated to their task. Therefore, we chose to identify the information in the "chart" by the name of the child and reworded help text to obviate mentions of charts. In order to minimize clutter in the interface associated with choosing among multiple charts, the initial design allowed users to set a chart inactive. In practice, the idea of archiving a chart turned out to be distasteful to parents. They felt it

would be a personal slight to the child. We were able to minimize the appearance of lists of children in the interface without including this feature.



Figure 3. Interaction flow for initial iPhone design.

Development of Web Designs

Tests of the web designs exposed similar problems to those found in testing the iPhone application. Users working with the web application had difficulty knowing for which child they were entering data. To solve this, we added more informative titles for data entry screens. The difficulty of knowing when their data had been saved was a problem, so we reworked the interaction flow to make use of modal dialog boxes.

One way in which the web site testing informed the iPhone interface was in the entry of sleep data. We found users were inclined to set and forget a timer for a long-duration event like sleeping, and the use of timers for sleep events was confusing. We chose instead, to collect a start time and a stop time for sleep events, cueing the user with a text based message on the top-level data entry screen when a sleep event was in progress.

Users were more insistent on similarity between the iPhone and web applications than we expected. One example is the configuration of application preferences for data entry and those for granting access to other caregivers. The two were initially grouped together on the web site but were separate on the iPhone. The iPhone link titles were necessarily shorter. Users looked for the same grouping of preferences and expected the same link titles in the two applications.



Annette Greiner March 11, 2010

Figure 4. Initial interaction flow for web application.

Final iPhone Design

The final design for the iPhone allows for much larger active button areas, especially on the data entry page. It also presents all the observation types with no scrolling. Through the design development, we found allowing a longer interaction path through the application often coincided with more clarity for the user. Since keeping the application easy for sleep-deprived parents to use was important, we felt a longer but more straightforward interaction was superior. (See Figure 5.)



Figure 5. Sample mockups for data entry screen and dialogs for final iPhone design. The landing page is much simpler and shorter than previous iterations. Also, sleep times are entered without a timer. The Breast Feeding link goes to a page with a timer or the option to enter times directly. The diaper screen gives the user a selection for dirty diaper colors that may also play a diagnostic role.

The screens for viewing the history of user entered data (Figure 6) became much more streamlined with successive iterations of the design. The final versions take advantage of detail disclosures that group what would otherwise be long lists.



Figure 6. History screens for final iPhone design. The user initially sees a list of months for which data has been captured. They can then drill down to a specific date to view observations grouped by type.

Screens related to milestones (Figure 7) are grouped by age ranges in which the milestones are expected to occur. The list of milestones was very popular with potential users in usability testing. They reported that they would enjoy looking through the list in order to determine what milestones their child would likely achieve next.



Figure 7. Final designs for milestone related screens for the iPhone application. Left is the top-level milestone page, showing age ranges for selection. Second from left is a list of milestones for one range. Right of that is a data entry dialog. Right shows the list modified after the user has entered a date for the first milestone.



Figure 8. Interaction flow for final iPhone design.

Final Web Design

The final design for the web application has strong and intentional similarities to the iPhone application. However, it takes advantage of the greater window size afforded on a computer screen. Figure 9 shows the final design of the KidOoDL home page.



Figure 9. KidOoDL.com home page before user login. New users can register immediately, returning users can readily log in.

Data entry on the web site is similar to data entry on the iPhone. The main differences is the use of standard web controls for selecting options and the use of modal dialog boxes layered over the referring page. Figure 10 shows the data entry page for the web application.

| O School of Information | | | |
|-----------------------------------|--|--|--|
| H 1 http://www.ischool.berkeley.e | du/ | RSS C Q+ protovis | |
| | - Home drop, to healthteam Google Docs | testuser@gmail.com • My Account • Help • Log Out Switch to Sammy V L OverAuthteredenzied to Discribit Non-Architel Anno Tor Attletion of Discribit | |
| Home | Milestones for Sammy | Last: imitated some movements and facial expressions Mar 1, 2010 | |
| Enter Data | Does Sammy enjoy playing with other people and perhaps o | ry when playing stops? | |
| History | Sleep for Sammy Enter New Sleep Event | Last: imitated some movements and facial expressions Mar 1, 2010 | |
| Reports Setup | Diaper for Sammy Enter New Diaper Event | Last: imitated some movements and facial expressions Mar 1, 2010 | |
| Caregivers About KidOoDL | Breast Feeding for Sammy Enter New Breast Feeding | Last: imitated some movements and facial Edit expressions Mar 1, 2010 | |
| | Bottle Feeding for Sammy Enter New Bottle Feeding | Last: imitated some movements and facial Edit expressions Mar 1, 2010 | |

Figure 10. Final design for the data entry page of the KidOoDL.com web application. Users begin an interaction flow for each observation type by clicking the corresponding button.



Figure 11. Final interaction flow for web application.

Visualizations

The web-based reports include charts and graphs that give users deeper insight into the data they entered. We designed four distinctly different views, two stand-alone views for viewing milestones, and two linked views for viewing daily observations. The visualizations were tested for usability with paper prototypes. Three parents were individually asked to answer a series of questions about "their" baby based on the graphs. We made major changes to all the visualizations based on their feedback.

Milestones

The milestones can be perused in a linear graph that plots a dot for each milestone the child has reached along a horizontal time axis organized by age ranges (Figure 12). The age ranges are color coded, so that dots occurring before or after their expected time appear in a different color than the surrounding dots. This makes spotting milestones outside their expected range easier. The milestones are grouped by category (e.g., social and emotional, movement, etc.) along a vertical axis. Milestones that have not occurred but were expected by the current time are shown in a far right column. The child's current age is indicated by a vertical line across all categories. When the page loads, the user sees

a compressed view showing only the dots in each category. Hovering the mouse over a dot reveals details such as a description of the milestone, the age range in which it is expected, and the date it was observed. Clicking in a category area expands that category to reveal the descriptions for each dot.



Developmental Milestones



We also developed a set of spider plots, also known as radar plots, of milestone data (Figure 13). These plots show concentric circles representing months of age. For each date range, a separate spider plot shows the range highlighted in the same color used in the linear visualization. Dots representing the milestones are each plotted, according to category, in one of seven 51-degree angular sections. The distance of each dot from the center corresponds to the date at which the milestone was observed. We connect the dots to get a shape extending beyond the shaded area if milestones have occurred late or extending inward from the shaded area if milestones occurred early. Rollover effects for each dot report the milestone description, date of observation, and expected age range.

CDC Developmental Milestones for: Sammy, 39 Months



Figure 13. Spider plots of milestone data. Concentric circles with a colored fill correspond to months in the given age range. Dots are plotted angularly by category with their distance from the center determined by the date of observation.

Observations of Daily Living

The daily observation data are plotted in a visualization that consists of a date range selector that shows an overview of all data entered as well as a detail plot. The detail plot shows either amount over the selected range of days (Figure 14) or events on a 24-hour schedule over the selected range of days (Figure 15). The date selector consists of sparkline-style linear plots of sleep, breast feeding, bottle feeding, and diaper amounts. A pair of sliders overlaid on the sparklines allows the user to select a range of time for detailed view beneath.



Figure 14. Amounts view for daily observations. At top is the date range selector showing sparklines for all entered data. Below is one week's worth of bar graphs with a hover state shown for one day's bottle feedings.

The version of the detail view that initially loads shows bar graphs for each of the four observation types for the previous week. The user can check a box to see the bars broken into individual naps. The user may also select whether naps that include the midnight hour are considered part of the previous day or broken into two naps for more accurate 24-hour totals. By clicking the Schedule button, the user can load a graph showing hours of the day (when naps occurred in time) along the vertical axis and days in the selected range along the horizontal axis. This view can be customized to show one or more observation types by clicking checkboxes.



Figure 15. Schedule view for daily observations for one week. The vertical axis represents time over a 24-hour period. The horizontal axis shows the dates selected with the range sliders (vertical blue bars). A hover state for a single nap is shown.

System Design

Platform Architecture



Figure 16. The platform architecture - complete workflow from parent to physician

KidOoDL is the first application built on the Berkeley Health Informatics (BHI) ODL Platform. This platform is our underlying architecture that supports ODLs and milestones of many types and from many sources, not just infant ODLs and not just KidOoDL. KidOoDL's workflow begins as a caregiver enters their infant observations and milestone tracking data on an iPhone. The data entry can also occur while there is no Internet connectivity. Web browser data entry is also possible via the KidOoDL web application interface, and the caregiver may use whichever mode is more convenient. Data entered on the iPhone can be synchronized with the web client (by pressing a button and using our RESTful² synchronization API (application programming interface), and data entered on the web client can be synchronized to the iPhone. Our team successfully designed and built a functioning synchronization engine.

Optionally, KidOoDL allows parents to send collected infant data for storage in personal health record (PHR) systems. Currently we connect with Google Health (via secure OAuth authentication) writing observations to a user-chosen Google Health profile.

² "REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. It encompasses the fundamental constraints upon components, connectors, and data that define the basis of the Web architecture, and thus the essence of its behavior as a network-based application." Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000. Web 6. May 2010

<<u>http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm#sec_5_2</u>>

Lucile Packard Children's Hospital at Stanford University currently uses the Cerner electronic medical record system (EMR), integrated with Google Health. Our system allows physicians to access KidOoDL data either through this interface, through their personal access to Google Health, and potentially through our web application. We also support parents sharing data with physicians by printing data via the web browser interface, or sharing information on their phone, directly.

Data Architecture



Figure 17: BHI ODL Platform Data Model

The data model was designed with flexibility and scalability in mind. It uses role-based access control (RGBAC) with four levels of access: Owner, Administrator, Contributor, Reader, and Physician. Owners can assign roles to others, Administrators have full access to a chart, but cannot revoke the owner privileges, Contributors can enter data on a chart, Readers have read-only access to the data, and physicians also have read-only access to the data.

The data model uses an abstract key-value representation for observations and milestones. Rather than defining a fixed set of observation types, this design allows new types to be represented in a standard way. Physicians can create new ODL types and package them in collections (called "panels") to have their patients track. A physician or hospital can also assign an existing panel of observations to collect.

Scalable Architecture



Figure 18: KidOoDL Infrastructure Diagram

The platform web service was implemented using the Pylons application framework after consideration of alternatives (see Appendix). We implemented our own Amazon Machine Image based on Ubuntu Linux. Our application currently runs in Amazon's EC2 with our development environment at WebFaction. We used the SQLAlchemy Object Relational Mapper (ORM) and a strict Model View Controller (MVC) Architecture.

Addendum

The Business Model

The management of personal health information is quickly becoming a classic two-sided market in which value is created for patients on one side and providers on the other. Health insurance and pharmaceutical companies and researchers (epidemiologists, etc...) also gain value, but they are secondary actors in the market. The development of personal health records (PHRs) and the push for digital health information exchange (HIE) have both hastened the evolution of this new market as the walls of institutional information silos across the country come crumbling down. However, because this market is new and evolving, the value propositions are still being developed, and companies remain uncertain about which side of the market to go after. Understanding the nuances of the market is imperative to developing a successful pricing and growth strategy. Our platform creates value for both patients and providers, giving us greater opportunity than many other offerings.

Patient Value Proposition

We provide the parents of newborns and small children with powerful tools to practice personalized preventative medicine. Using a combination of individual data, aggregated user data, established pediatric guidelines, and translational medicine, our system facilitates the discovery of personalized normal ranges for a variety of common conditions. The system also has advanced developmental tracking capabilities enabling parents to follow the growth of their child across 7 different categories.

Our system has lessened the burden of data entry through careful user interaction research and support for multiple devices, platforms, and caregivers. We've also developed a creative reward system that makes health tracking more fun and interesting.

We understand that ultimately health is a social endeavor. We support both self-controlled, but personally identifiable, and anonymous sharing of selected health data. Sharing can happen in a number of ways including with other users in our network as well as through popular social networks such as Twitter and Facebook. We provide tools for connecting users with similar characteristics for sharing within our own network.

Ultimately, our goal is to empower parents with the best tools for managing their children's health.

Provider Value Proposition

Providers can realize substantial benefits from using the new patient communication channel our platform enables. Creating additional touch points between patients and physicians beyond the intermittent clinical visits creates practice loyalty and significant brand equity. By allowing the provider to actively engage the patient in his or her own healthcare process the platform improves the experience for both parties.

The provider is additionally incentivized by the delegation of data collection duties to the patient. Portions of the data required for CMS billing as well as data recommended by the American Association of Pediatricians (AAP), all of which has traditionally been collected in clinic, can now be easily entered by the patient from home prior to their visit. In some cases, developmental milestones being the prime example, our platform may significantly improve adherence to recommended practice, even helping in educating the pediatrician. In all cases time is freed up for the more valuable, personable, and nuanced interactions that should shape a clinic visit.

We also provide advanced data visualizations that enable more efficient consumption of patient histories by the pediatrician. A provider can glance at trend lines and quickly determine whether there are any specific areas that need addressing.

Pricing & Growth Strategy

The fact that we are a platform in a two-sided market means we must consider network effects in determining our pricing and growth strategy. For example, a provider is much more likely to use our platform if a significant portion of their patients are using it. Conversely, a patient is highly incentivized to choose our platform if their provider supports it. Because of the existence of network effects it will be important for us to employ a strategy that maximizes the growth of our user base.

In order to quickly gain patient adoption we will provide them free access to our data collection tools including our iPhone application. We will additionally provide them the ability to print out or export their data for free. However in order for patients to access value added tools such as visualizations and networking, or enable integration with their provider, we will charge a monthly subscription fee.

Because of powerful network effects exhibited by provider adoption, we've elected to give them free access to our platform. Furthermore, we will implement additional incentives for them to encourage their patients to use the platform. These may include discounted subscriptions to all of their patients, or access to additional value added tools and services.

Appendix I

Application Framework Decision Analysis

SWOT Analysis I Regarding the decision to build using Pylons + SQL Alchemy instead of Django with its native ORM as the application framework

| Strengths | Weaknesses |
|--|---|
| Strengths Built for superior database functionality with native SQLAlchemy integration. (SQL Alchemy's Data mapper pattern is more flexible than Django's active record pattern). We know a good tactical hosting site that keeps up-to-date with pylons packages/security updates to get us started quickly before we move to a cloud-based solution. A more customizable, less-brittle, and more modular architecture than Django. You are | Weaknesses A much steeper learning curve for our team than with Django. Very small developer / user / support community compared to Django (order of 35:1) Fewer code samples available than for Django. Cannot plug-in entire 3rd party application stacks like you can with Django, so we'll have to build everything 'from scratch' which may be time consuming. This could be especially challenging with oAuth / Google Health integration. |
| Fewer "Magic" functions than Django. You are more in-control of your application than with Django. | integration. |
| | |
| Opportunities | Threats |
| Able to get assistance from iSchoolers if we run into trouble. (something we did not end up using). | Project more-likely to be abandoned than Django. Modules may not be kept up-to-date as well as |
| Can to leverage parts of a third-party open- source Pylons application for email validation. | Django. Potentially we will experience slower |
| More long-term flexibility. | development than with Django. |
| SqlAlchemy supports database sharding (horizontal partitioning with sessions) which we may use for database scaling. | |

SWOT Analysis II

(An analysis of strengths, weaknesses, opportunities and threats) regarding the decision to build using the Django + SQL Alchemy application frameworks combined, instead of building with the Pylons + SQL Alchemy application framework

| Strengths | Weaknesses |
|--|--|
| Django has a significantly larger developer community than the Pylons developer community. As a result, it has much better and more-through documentation and support than Pylons. Two of our team members have experience developing using Django but no one has any Pylons experience. Django includes user management / authentication built-in, but it's only useful if we can figure out interoperation between the Django and SQLAlchemy ORM. | Using SQLAlchemy with Django, instead of the native Django ORM, we lose many useful Django features (such as the automatic administrative interface), and we may subject ourselves to unknown integration complications if we use both ORMs. Django will never treat SQLAlchemy as a first- class citizen. Mixing the two ORMs creates a bit of a Frankenstein, combining the applications in a hackish, un-natural fashion. We do not have experience developing with Django + SQLAlchemy . Replacing the ORM is reputedly onerous: (http://lethain.com/entry/2008/jul/23/replaci ng-django-s-orm-with-sqlalchemy/). Only a small subset of the Django community uses this method. Django does not accommodate "out of the stack" components as well as Pylons. Django has a less-robust templating engine than Pylons . (minor) |
| Opportunities | Threats |
| Django has useful libraries/plugins that we could use for our project (Pylons largely lacks these in comparison) (e.g.: for email authentication: <u>http://github.com/redvasily/django- emailauth</u>). Using these plugins could vastly speed our development time (e.g.: for email authentication). | Uncertainty over the learning curve and difficulty of integrating Django with SQLAIchemy. Uncertainty over the robustness of using Django and SQLAIchemy. |
| | |

| continue to be larger than Pylons'. Community | |
|---|--|
| in the future. | |
| Django's functionality will improve rapidly. | |

SWOT Analysis III

Regarding the decision to build using Django with its native ORM (Object-Relational Mapper) and the SQL Alchemy application framework instead of building with the Pylons + SQL Alchemy application framework

| Strengths | Weaknesses |
|--|--|
| Administrative interfaces can be automatically be generated if you use the native ORM. Faster initial development time. Two of our developers already have Django experience. Superior plugin ecosystem. Django has a significantly larger developer community than the Pylons developer community. As a result, it has much better and more-through documentation and support than Pylons. Two of our team members have experience developing using Django but no one has any Pylons experience. | Django's native ORM is not as robust as SQLAlchemy's. If we want a complex data model we may have to resort to raw sql queries in our codebase which can make code maintenance and security-exploit scrubbing more cumbersome. Less customizable than a Pylons-SQLAlchemy solution. Django 's functionality is frequently more "magic" (less explicit and less visible) than Pylons (where you are more-able to work explicitly, with the details of implementation). |
| Opportunities | Threats |
| Django has useful libraries/plugins that we could use for our project (Pylons largely lacks these in comparison) (e.g.: for email authentication: <u>http://github.com/redvasily/django-</u> <u>emailauth</u>). Using these plugins could vastly speed our development time. | Django + SQL Alchemy 's support for database sharding (partitioning) (useful for database scaling) would be more difficult to implement than if building with Pylons + SQL Alchemy. |

Web Framework Decision: Pylons

We decided to go with Pylons as our application framework because, though it may take more development time, it should give us more flexibility and control and this aligns with our longer-term goals of developing a modular service. Pylons' superior database features and better customizability should serve this end well, though we will miss the wider support base and code samples of the Django ORM (Object-Relational Mapper). Since SQLAlchemy will never be a first-class Django citizen, the risk of it not properly functioning is too great for us to use it in our project. By using SqlAlchemy, we would lose many of the benefits of using Django, anyway.