Feast

Alive Without Compromise

Daniel Shiferaw
Software Engineering

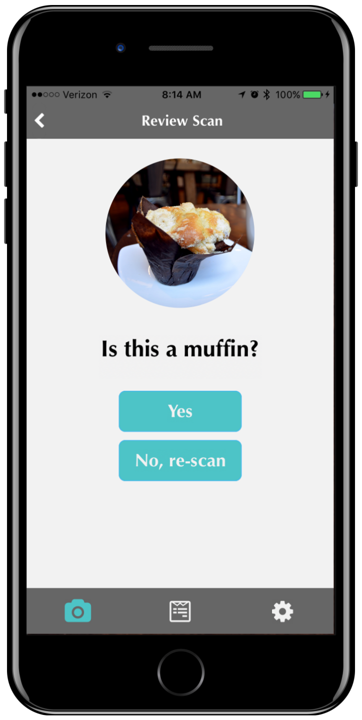
Peter Lowe
Design and Usability

Megan Wilson
Product and Team Manager

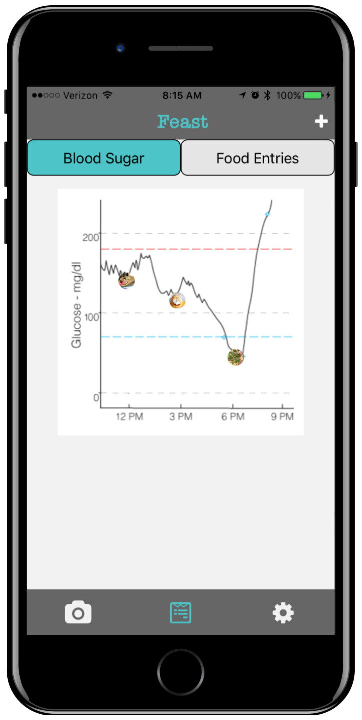
Problem and Solution Overview

Young diabetics often struggle with incorporating health limitations into their identity. This struggle is brought to a head around food because eating out is often a significant portion of social time and personal expression. Accurate sugar and carbohydrate data is often unavailable, which forces diabetics to choose between prioritizing their health and "having a life," which means both sharing and documenting their experiences. Our team seeks to enable diabetics to embrace unfamiliar cuisine and socialize around food while caring for their health by helping them calculate nutritional information and visualize health trends.

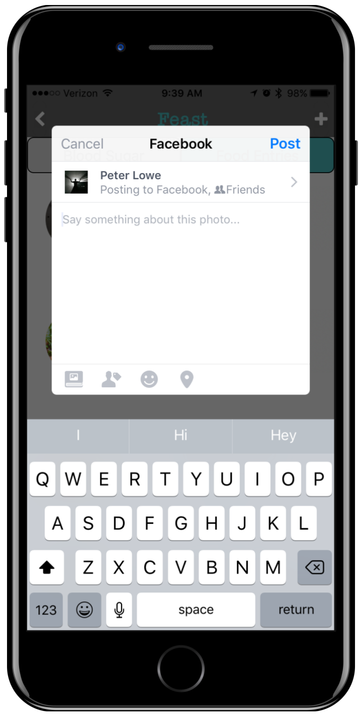
The goal of *Feast* is to help diabetics prevent their dietary and health goals from interfering with them living the life they want to live. The way *Feast* seeks to accomplish this goal is by streamlining the way diabetics obtain nutrition information about their food, keep track of their health statistics over time, and share their food experiences with others. *Feast* allows users to input the food they are eating in one of two ways: by using their phone camera to obtain a three-dimensional scan of their food, or by selecting the food they are eating from a database of options. *Feast* then uses the scan or food indicated to generate nutrition data about the food so that the user can add this data to their health log within the app, which also integrates with CGMs (continuous glucose monitors) to allow easy visual comparison of blood sugar trends against food eaten. Finally, the app allows users to share an image of their food through a variety of social media outlets. This feature seeks to reduce the hassle and hesitation diabetics may face when sharing their food digitally - an activity that has become a phenomenon among young social media users - and to encourage diabetics to be proud of the effort they devote to managing their illness rather than hiding it from their families and friends.



Food recognition



Logging blood sugar data and food eaten



Sharing food on social media

Design overview with 3 user tasks

Task and Final Interface Scenarios

Feast is designed to support users primarily across the following three tasks: 1) obtain nutrition information about their food, 2) keep track of measures of health over time, and 3) share food with others.

In talking to diabetics, our team discovered that obtaining relevant nutrition information about food (mainly total carbohydrates) is particularly difficult when eating out. As such, we decided to focus our app on reducing the friction in obtaining accurate information. We felt that designing *Feast* to extract nutrition information from a quick scan of food taken through a smartphone would greatly reduce the burden associated with eating out as a diabetic. We also designed our app so that, in the case where the user does not feel that taking a 3D scan of their food is appropriate, they can access nutrition information about the item they ordered using our database of foods. Figure 1.1 below demonstrates the task of obtaining nutrition information about a food using the 3D scanning method and figure 1.2 shows the the database lookup method, respectively.

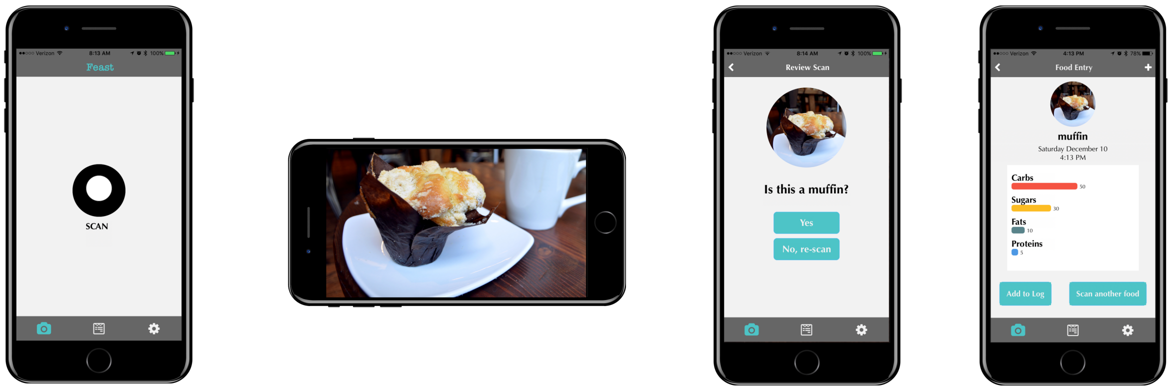


Figure 1.1: scanning process

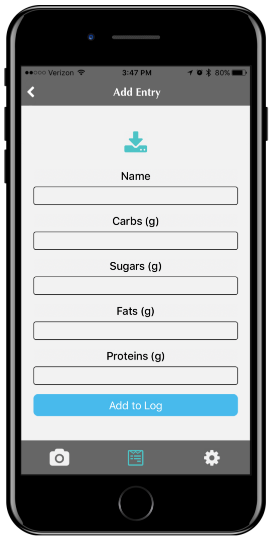
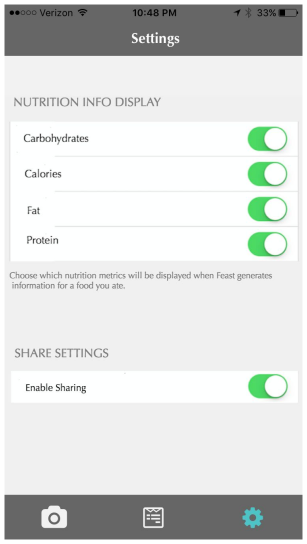


Figure 1.2: database lookup

Our needfinding interviews also revealed that it is difficult for diabetics to keep track of their health data over time. For example, it is difficult for diabetics to track trends in their blood sugar levels and calorie consumption over a period of several days, weeks, or months. *Feast* helps diabetics accomplish this task by storing and synthesizing the nutrition information of the foods a user eats and generating graphical representations of this data to allow for easy analysis of these metrics over time. Given that diabetics have customized health needs, users can configure *Feast* to keep track of and display the nutrition data that is most relevant them. This is demonstrated in figure 2.1 below, which shows the Settings screen of the app. Figure 2.2 shows the process of adding an item to the user's nutrition log after obtaining nutrition data by way of task 1. Figure 2.3 shows the graphical representation option and table view option with which the users can select to view their nutrition data. (Note: the user is free to toggle between these representation options by selecting "Blood Sugar Graph" or "Food Entry Table.")



Settings

NUTRITION INFO DISPLAY

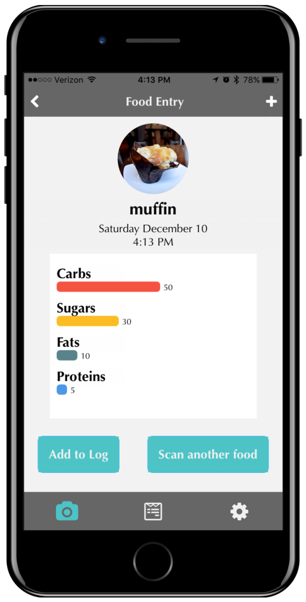
Carbohydrates	✓
Calories	✓
Fat	✓
Protein	✓

Choose which nutrition metrics will be displayed when Feast generates information for a food you ate.

SHARE SETTINGS

Enable Sharing	✓

Figure 2.1: settings screen


Verizon 4:13 PM 78%

Food Entry

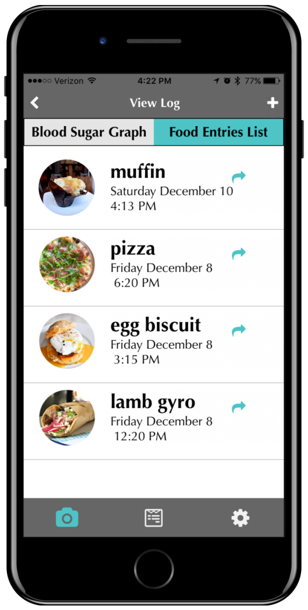
muffin

Saturday December 10
4:13 PM

Carbs	50
Sugars	30
Fats	10
Proteins	5

Add to Log

Scan another food


Verizon 4:22 PM 77%

View Log

Blood Sugar Graph | Food Entries List

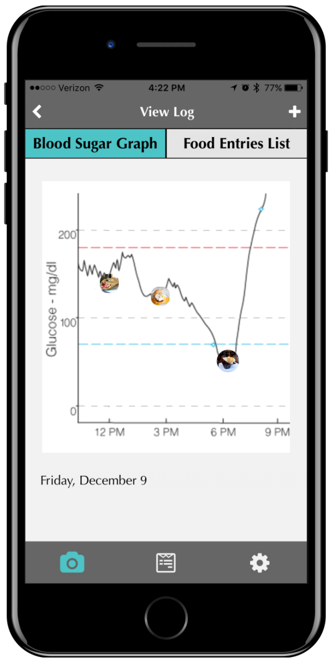
muffin Saturday December 10 4:13 PM

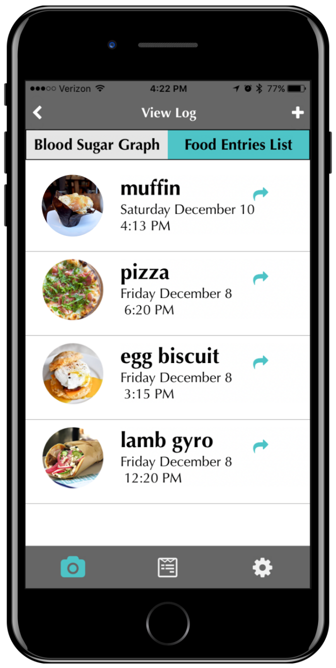
pizza Friday December 8 6:20 PM

egg biscuit Friday December 8 3:15 PM

lamb gyro Friday December 8 12:20 PM

Figure 2.2: Adding food to log via scanning




Verizon 4:22 PM 77%

View Log

Blood Sugar Graph | Food Entries List

muffin Saturday December 10 4:13 PM

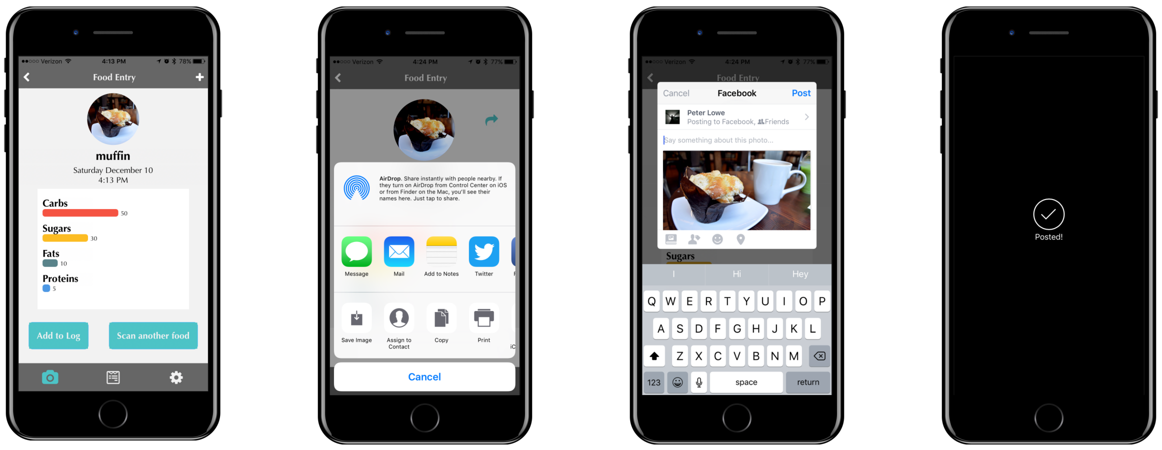
pizza Friday December 8 6:20 PM

egg biscuit Friday December 8 3:15 PM

lamb gyro Friday December 8 12:20 PM

Figure 2.3: Graphical and list view of log

The third main task of our app is sharing experiences around food. Here we refer to "sharing" in both the digital and physical senses, however the implementation of this task focuses primarily on digital sharing. *Feast* facilitates users sharing the 3D scan of their food generated during task 1 on several social media outlets including Facebook, Instagram, Twitter and LinkedIn. Figure 3.1 represents the process of completing this share on the app. Our team also feels that this task of supporting diabetics in sharing experiences around food is accomplished across all of the three tasks. As mentioned in our mission statement above, our main goal is to reduce the obstacles that diabetics face when eating out. We feel that if our three tasks accomplish this goal, then we are making it easier for diabetics to engage with their friends and families in social situations involving food.



AirDrop. Share instantly with people nearby. If they turn on AirDrop from Control Center on iOS or from Finder on the Mac, you'll see their names here. Just tap to share.

Figure 3.1: Sharing food on social media

Design Evolution:

In this section we will show the design evolution of *Feast* from our initial sketches to through our medium-fidelity prototype.

After conducting needfinding interviews, generating empathy maps and narrowing down our point of view and "how might we" questions to guide our project, we generated several experience prototypes to gain a better understanding of how we might help diabetics reduce the impact of their illness on their ability to embrace life experiences. Two of these experience prototypes, *Adventure Life* and *Blue Apron + Chronic Illnesses* largely inspired *Feast*. The idea behind *Adventure Life* was to help diabetics find challenges that do not violate the constraints of their illness, set goals and track progress toward these goals. Our sketch from *Adventure Life* is shown in Figure 4.1 below. *Blue Apron + Chronic Illnesses* was designed to encourage diabetics (and individuals suffering from other chronic illnesses requiring dietary restrictions) to explore new foods using meal preparation kits designed with their needs in mind. Our sketch from *Blue Apron* + *Chronic Illnesses* is shown in Figure 4.2 below.

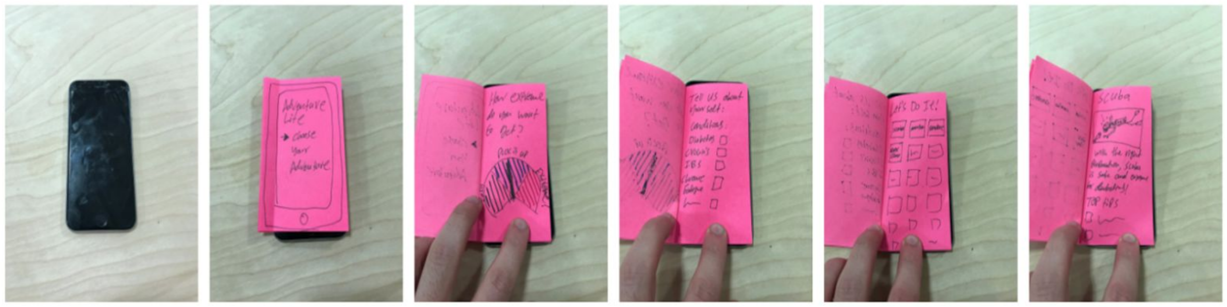


Figure 4.1: Sketches of *Adventure Life* experience prototype.

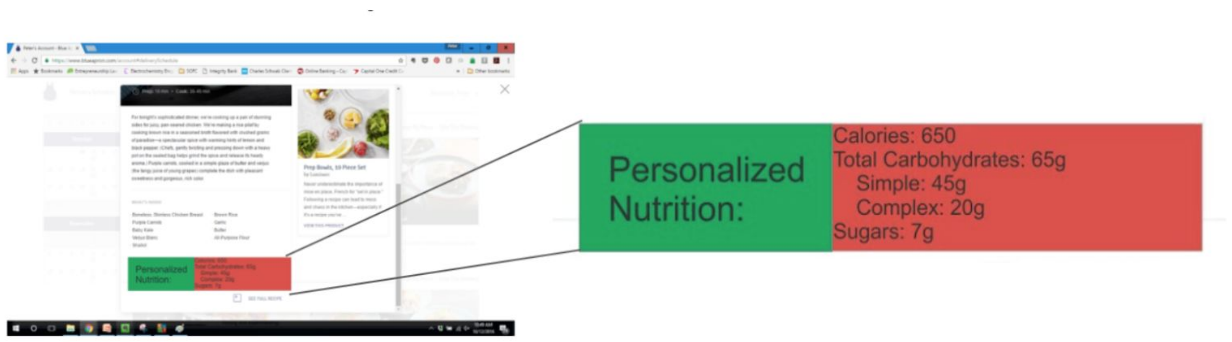


Figure 4.2: Diagram detailing *Blue Apron* + *Chronic Illnesses* experience prototype.

Feast was inspired by the "carpe diem" attitude of *Adventure Life* and the nutrition aspect of *Blue Apron* + *Chronic Illnesses*. We received feedback from our experience prototype testing that *Adventure Life* would be largely geared toward adventure-apt diabetics who may seek these challenges on their own, and that *Blue Apron* + *Chronic Illnesses* would appeal to the subset of people suffering from chronic illnesses who have the time and interest to devote to cooking. Our group recognized that pursuing an idea like *Feast* which encourages diabetics to be unafraid of trying new things in social settings involving food - a situation that most individuals find themselves in at one point or another - would allow us to reach (and hopefully impact) a broader audience.

After settling on the idea to develop a product to make eating out easier for diabetics, we created a concept video for *Feast*. In this video, we show a diabetic going out with his friends before he has the *Feast* app and suffering from an elevated blood sugar afterwards. We then show this same diabetic going out with his friends once he is able to obtain the nutrition information of his food through *Feast*, and we see him enjoying himself without suffering health consequences at the end of the night.

Once we refined our concept in our concept video, we began considering the actual implementation of *Feast*. Our team discussed the advantages and drawbacks to creating *Feast* with an Apple Watch interface and creating *Feast* with a traditional smartphone interface. After analyzing both options, we chose the smartphone option, again opting to maximize the pool of users our app could potentially benefit. Figure 4.3 shows our sketches of the Apple Watch interface and Figure 4.4 shows our initial sketch for the *Feast* smartphone interface. Figures 4.5, 4.6, and 4.7 represent the three *Feast* tasks of obtaining nutrition information about a food, keeping track of nutrition data over time, and sharing food in some of our early sketches using the smartphone interface.

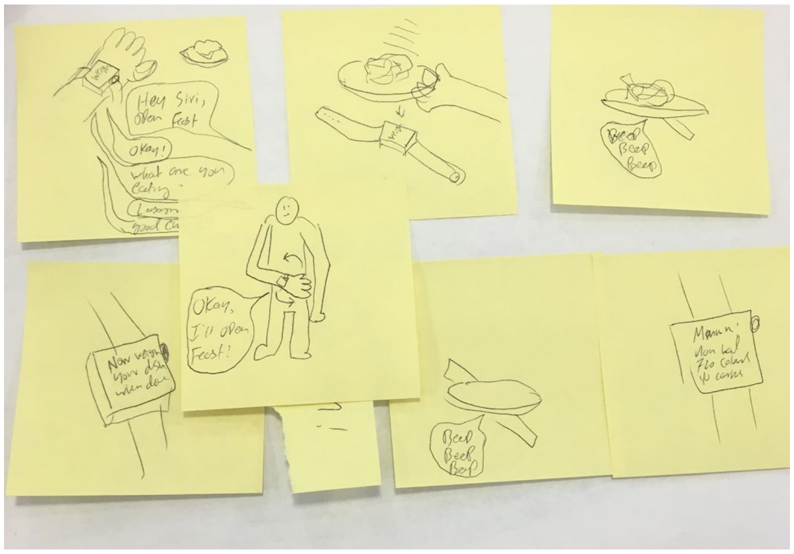


Figure 4.3: Sketches of the *Feast* concept using an Apple Watch interface. This idea featured voice commands and gestural communication with the application.

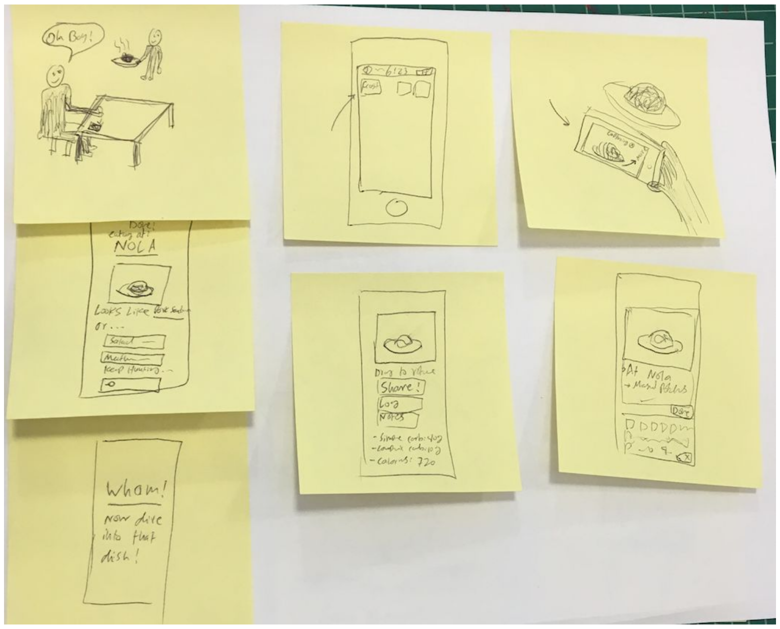


Figure 4.4: Initial storyboard of the smartphone interface implementation of *Feast*.

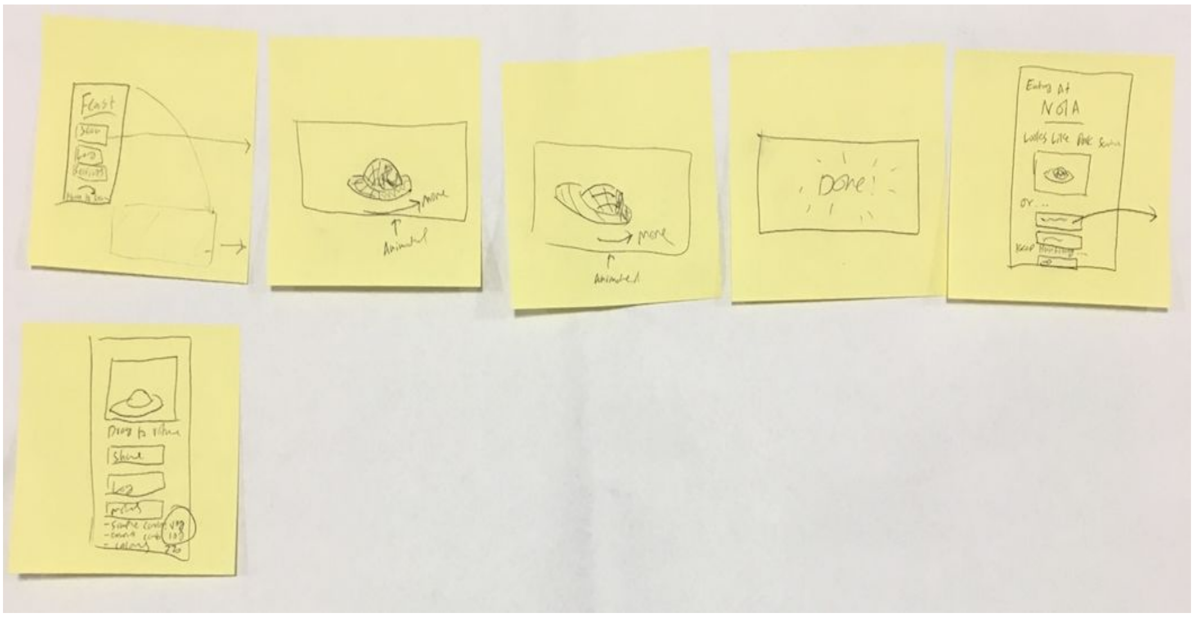


Figure 4.5: Original task flow of obtaining nutrition information about a food using the 3D scanning feature of the smartphone interface. (Note: in this phase of the design process, we had not yet fleshed out the custom entry option for obtaining nutrition information about a food.)

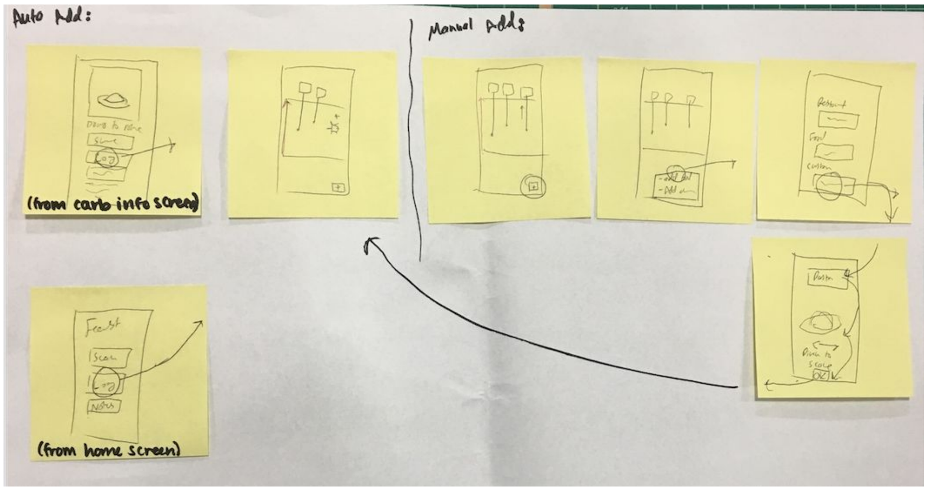


Figure 4.6: Original task flow for keeping track of nutrition and health data over time. This task flow shows how a user would view their log immediately after adding an item to the log, after manually adding an item (rather than adding it using the 3D scanning functionality), and from the *Feast* home screen.

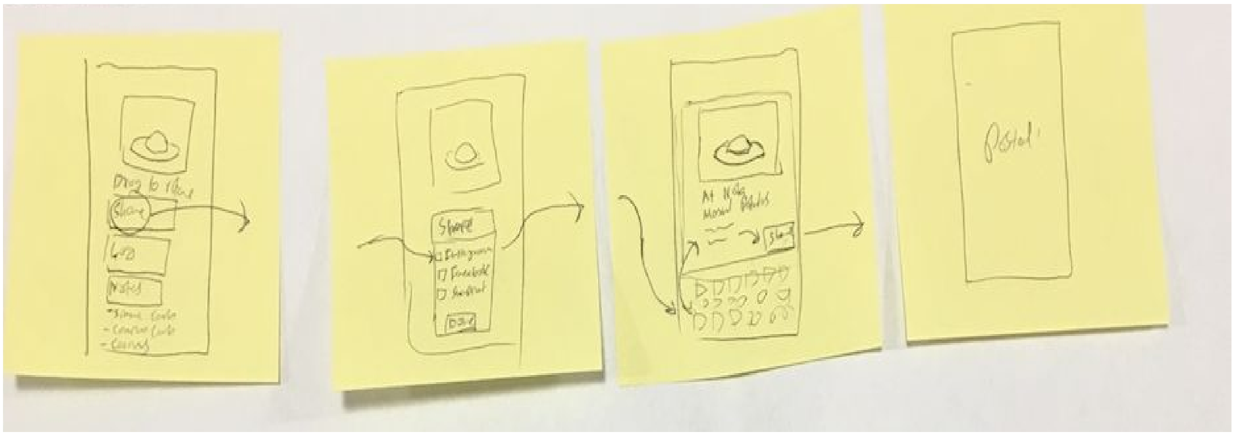


Figure 4.7: Early task flow for sharing food from the *Feast* app to social media networks.

Once we diagrammed our vision for how each of the three tasks might look on *Feast*, we constructed our low-fidelity prototype from the above storyboards. Figure 4.8 represents task 1 in our low-fidelity prototype, Figure 4.9 represents task 2 in our low-fidelity prototype, and Figure 4.10 represents task 3 in our low-fidelity prototype.

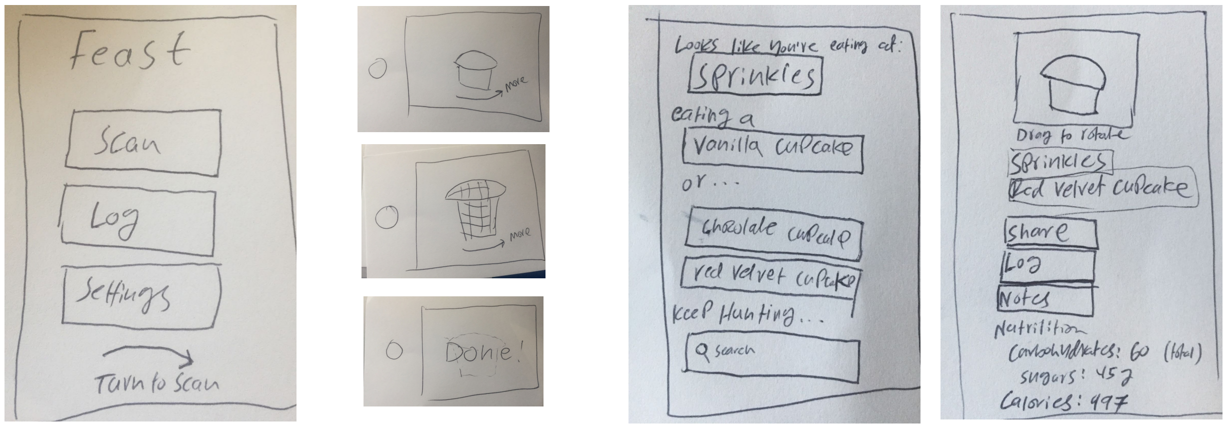


Figure 4.8: Scanning in low-fi prototype

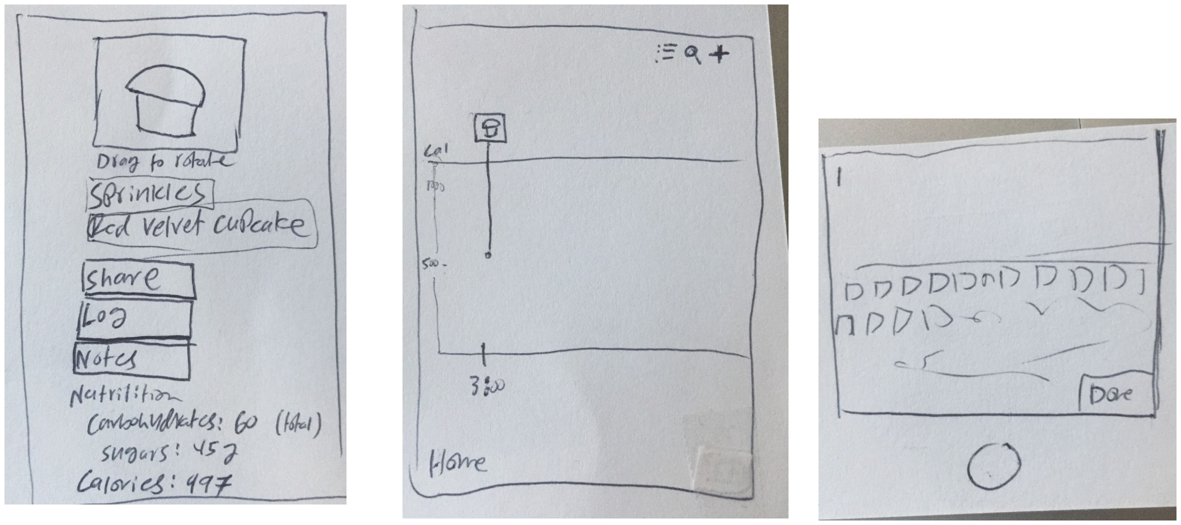
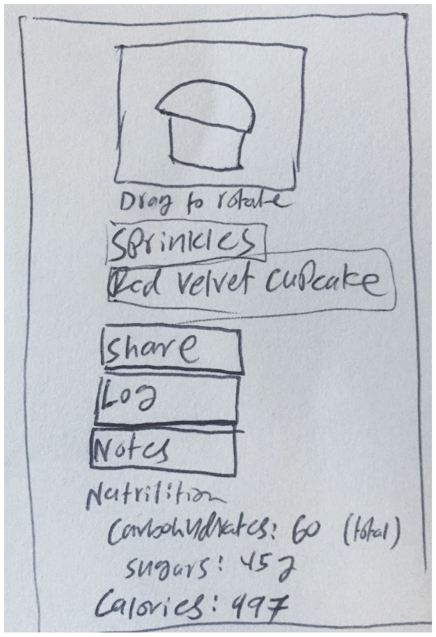


Figure 4.9: logging in low-fi prototype



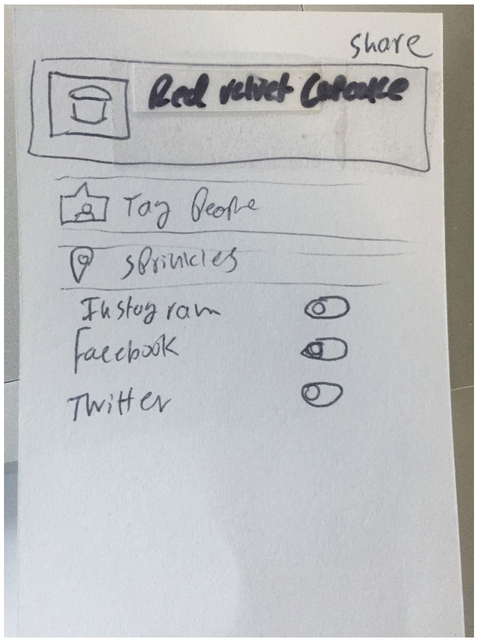


Figure 4.10: sharing in low-fi prototype

Testing our low-fidelity prototype yielded several pieces of feedback on our implementation of the three tasks. In task 1, users felt that the instructions to rotate their smartphone to the landscape orientation to trigger scanning were not intuitive, they had difficulty understanding what to do to obtain the “scan” of their food based on the design of our prototype, and noted that the nutrition information the app was generating for them was not displayed front and center on the interface. In task 2, users liked the idea of side-to-side scrolling over time on the log, but felt that it needed to be more personalized. Finally, in task 3, users remarked that the straightforward flow of sharing the scans of their food to social media sites was great, but argued that many diabetics wouldn’t use this functionality.

We revised our design idea and constructed our medium-fi prototype with these pieces of feedback in mind. We updated task 1 to include scanning instructions, added a button to trigger the scan for users who were less comfortable with the shortcut of rotating their phone to the landscape position to trigger the scan, and moved the nutrition data to the center of the screen when it is displayed for the user. Our team updated task 2 to flesh out the custom/manual add functionality of the app and allow users to indicate the portion size of the meals they add manually using a pinch-to-zoom technique. We did not make any significant changes to the task flow associated with task 3 and decided to use our medium-fidelity prototype testing to delve more into the sentiment of our low-fidelity prototype testers that diabetics do not frequently share their food on social media. Figure 4.11 represents the task flow associated with task 1 in our medium-fidelity prototype, Figure 4.12 represents the task flow associated with task 2, and 4.13 represents the task flow associated with task 3.

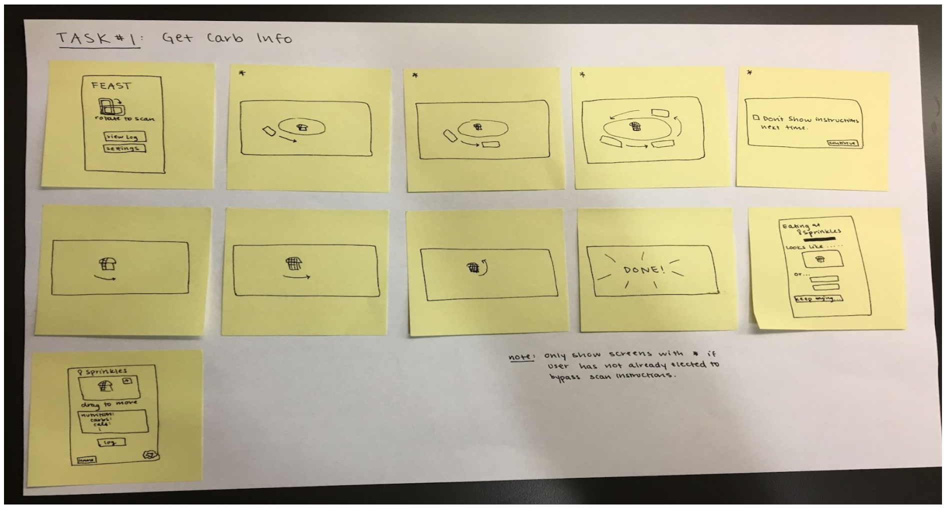


Figure 4.11: Task flow for obtaining nutrition information for the medium-fidelity prototype of *Feast.*

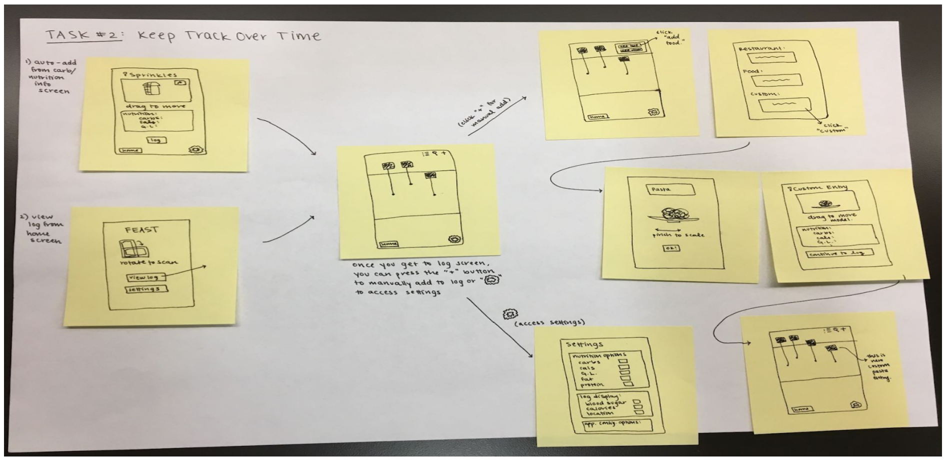

TASK #2: Keep Track Over Time
1) auto-add from carb/nutrition info screen
2) view log from home screen
Sprinkles
drag to move
nutrition: carbs: cals: G.L.:
log
home
FEAST
rotate to scan
view log
settings
once you get to log screen, you can press the "+" button to manually add to log or "⚙" to access settings
(click "+" for manual add)
click "add food."
Restaurant:
Food:
Custom:
click "custom"
Pasta
pinch to scale
ok!
Custom Entry
drag to move model.
nutrition: carbs: cals: G.L.:
continue to log
(access settings)
Settings
nutrition options
carbs
cals
G.L.
fat
protein
log display:
blood sugar
calories
location
app. config. options:
this is new custom pasta entry.

Figure 4.12: Task flow for logging nutrition data over time in the medium-fidelity prototype of *Feast*.

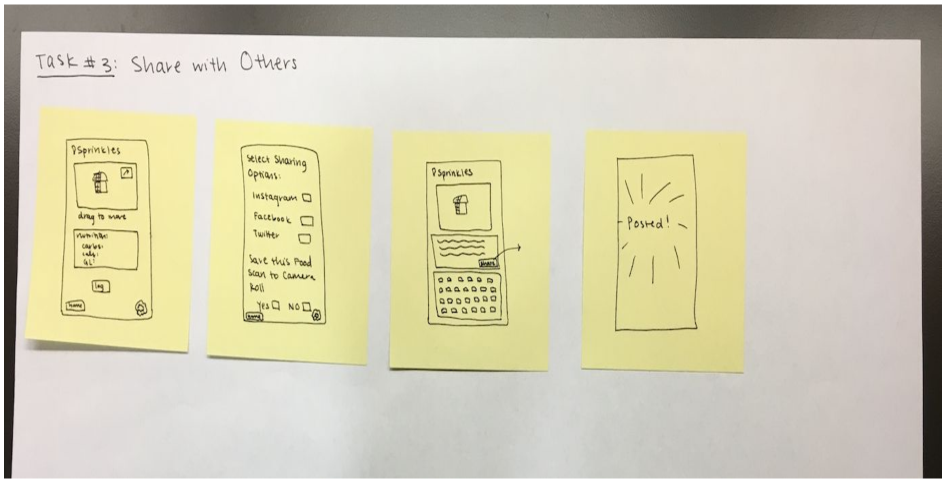
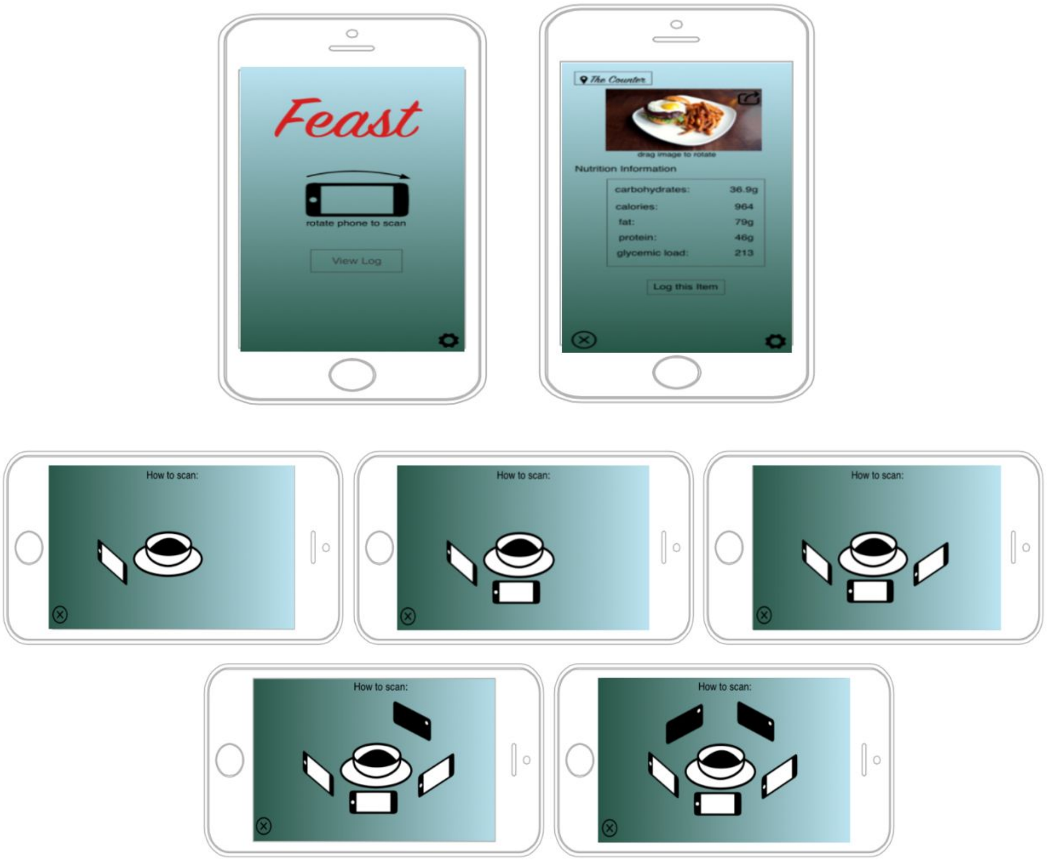


Figure 4.13: Task flow for sharing food from *Feast* to social media sites in our medium-fidelity prototype.

The figures below show the three tasks in our medium-fi prototype.



carbohydrates:	36.9g
calories:	964
fat:	79g
protein:	46g
glycemic load:	213

Figure 4.14: Screens from task 1 in our medium-fidelity prototype.

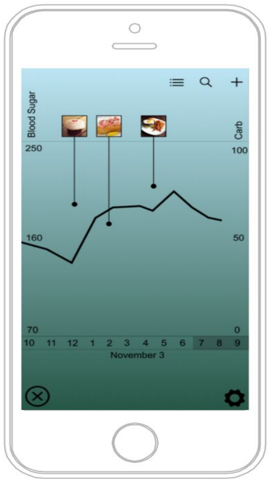


Figure 4.15: Log screen from medium-fidelity prototype.

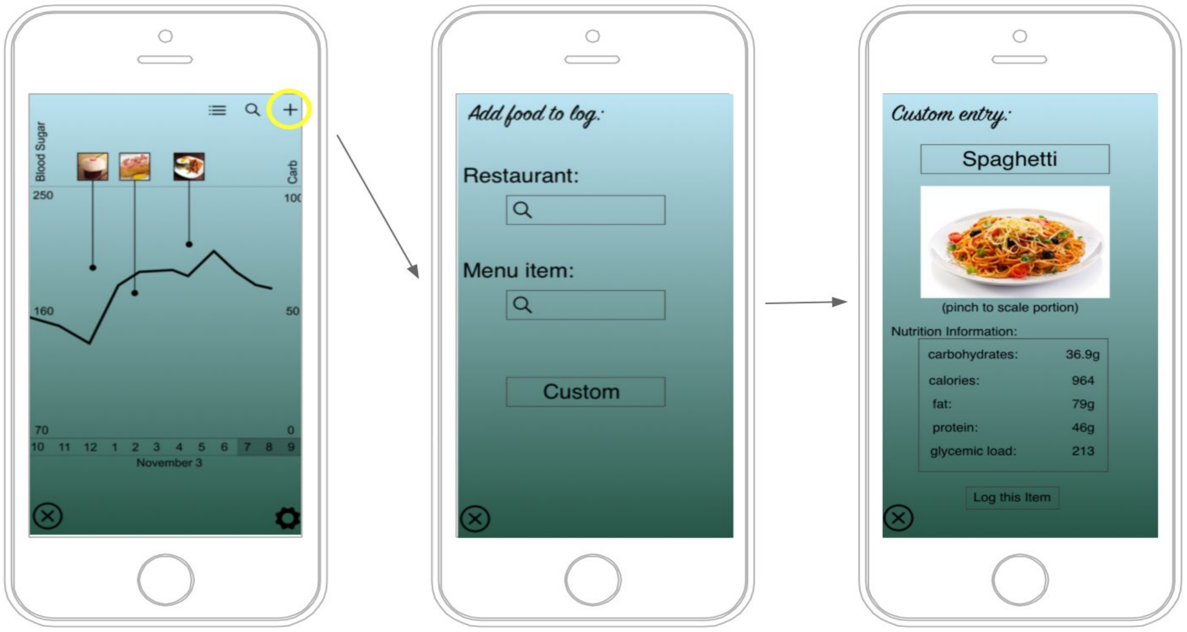
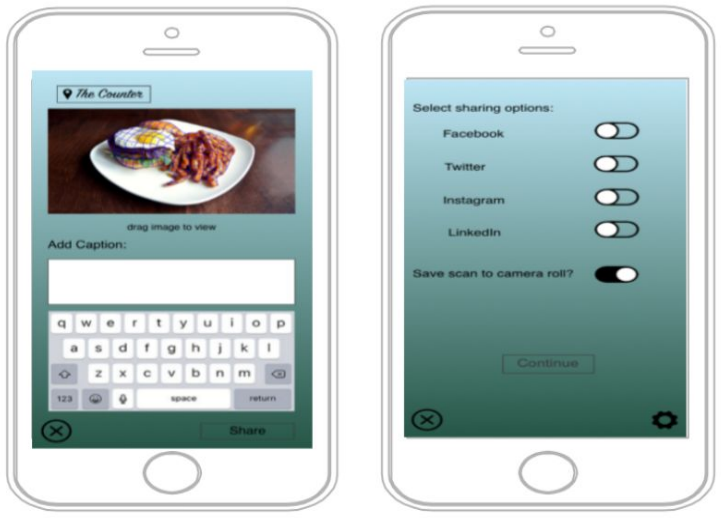


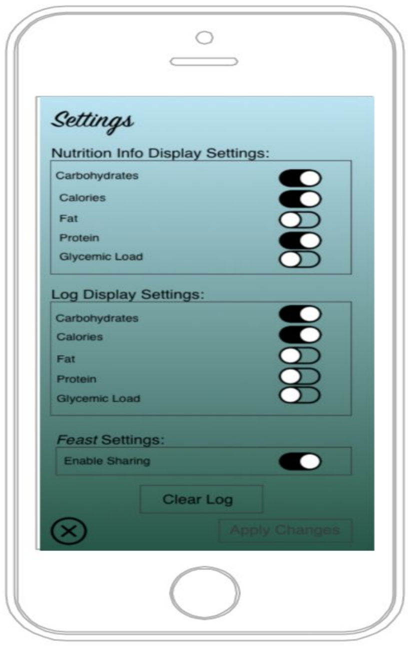
Figure 4.16: Task 2 Task flow in medium-fidelity prototype.



Select sharing options:

Facebook	○
Twitter	○
Instagram	○
LinkedIn	○
Save scan to camera roll?	●

Figure 4.17: Screens from task 3 in our medium-fidelity prototype.



Nutrition Info Display Settings:

Carbohydrates	●
Calories	●
Fat	○
Protein	●
Glycemic Load	○

Log Display Settings:

Carbohydrates	●
Calories	●
Fat	○
Protein	○
Glycemic Load	○

Feast Settings:

Enable Sharing	●

Figure 4.18: Settings screen from our medium-fidelity prototype.

Major Usability Problems Addressed

In this section we will show the design evolution of the medium-fidelity prototype of *Feast* to the high-fidelity prototype of *Feast*. Our medium-fidelity prototype was tested using a heuristic evaluation, so this section will focus largely on the feedback we received from our evaluators.

The suggestions we received from our heuristic evaluators included grouping related tasks and streamlining the navigation from task to task, standardizing fonts and design elements across the app, refining the “share” task to address concerns related to security and industry standards, and including unit measurements to make data more meaningful to users. We considered the heuristic evaluators’ feedback in decreasing order of severity, focusing our efforts on revising the design and usability issues that most significantly inhibited our users from completing our three main tasks.

Many of our evaluators indicated that issues with the visual design of our app complicated the functionality unnecessarily. First, they cited that the gradient design of the screens in our medium-fi prototype made several of the buttons hard to read. After experimenting with lightening the gradient to make buttons on the screen more visible, our group decided to remove the gradient altogether; it seemed to present a disproportionate amount of usability problems compared to the relatively low benefit it gave our app in terms of visual design. We also received the feedback that our medium-fi prototype lacked consistency in the buttons used to navigate back to a previous screen or to return to the home screen. Evaluators noted that, because of these inconsistencies, they found themselves having to redo the entire scanning process after making a single mis-click. To address these concerns, we restructured the navigation within our app using two design paradigms common in iOS design. First, we created a navigation bar that appears at the bottom of each screen. This allows users to quickly access our two most important tasks (the camera icon facilitates scanning and clicking the log icon allows users to view their log instantaneously) as well as the settings options, which largely dictate the nature of their interactions with *Feast*. Secondly, we implemented a simple carrot-shaped back button in the top left corner of each screen. This allows users to navigate back one screen, and we expect this design to be intuitive to most users as it is a widely-used feature across iOS apps.

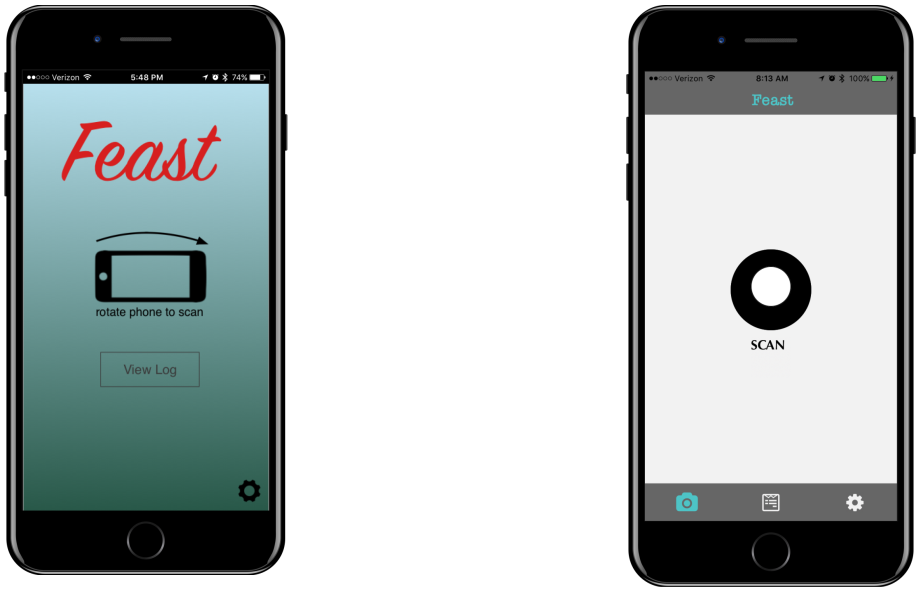


Figure 4.19: Medium and hi-fi home screens and navitation

Another high-severity problem that our evaluators identified was the lack of axis and unit labeling throughout the app. Without units, the nutrition data that *Feast* generates for users is largely useless. To address this concern, we redesigned our log to include axis labels on our graph to indicate that blood sugar is being logged in mg/dL. We also included an x-axis label that indicates the date and time so that users can easily identify the times at which their blood sugar spiked and plummeted. The style of x-axis labeling in our graph was inspired by the charts in the Apple Health app, as we feel that the design of Apple Health is informative yet minimalistic.

In addition to the units on the log being unclear, evaluators indicated that it was difficult to indicate the portion size of the food they were adding using the manual entry function. This is because the manual entry function in our medium-fi prototype asked users to indicate portion size by pinching-and-zooming on a stock image of the food that did not contain any reference points. We initially toyed with the idea of adding a reference point to the stock images for manual data entries to make the pinch-to-zoom portion-size indication work, however it quickly became clear that this strategy would result in a more cluttered and still-not-intuitive design. Our group also considered using a pie chart to allow users to indicate portion size, but we felt that the lack of reference points issue was still present in this possible solution. In the end, we decided to have users indicate portion size in a more generic way: in traditional volume and weight measurements. For example, a user would be able to input the amount of rice in cups or the amount of chicken in ounces. While this input entry does not include any novel volume measurement tools, it is simple, straightforward, and consistent with the way food is traditionally measured. Especially given that most diabetics have lived a lifetime of measuring portion sizes to factor into their blood sugar calculations, and since many popular nutrition apps use this model, we felt that this simple entry method for portion size would streamline the process of adding a manual entry from our implementation in our medium-fi prototype. Aside from the confusing pinch-to-zoom portion indication tool, we received feedback that it was unclear what several of the input boxes on the manual entry screen were intended for. To address this concern, we added straightforward labels (with units!) to the manual entry search screen to make it clear which data should go in which box. We believe that this simple fix will greatly improve the user experience for manually adding an entry to the log.

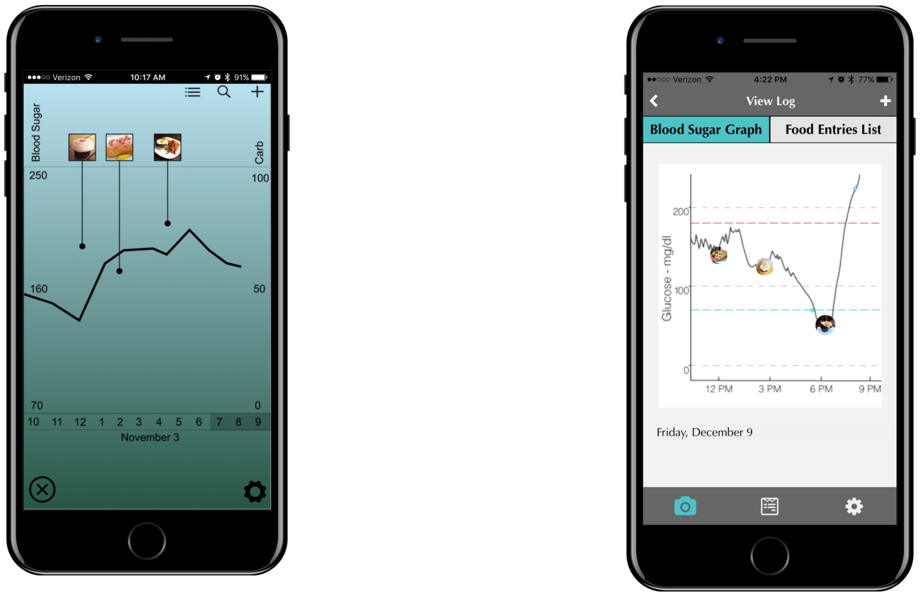
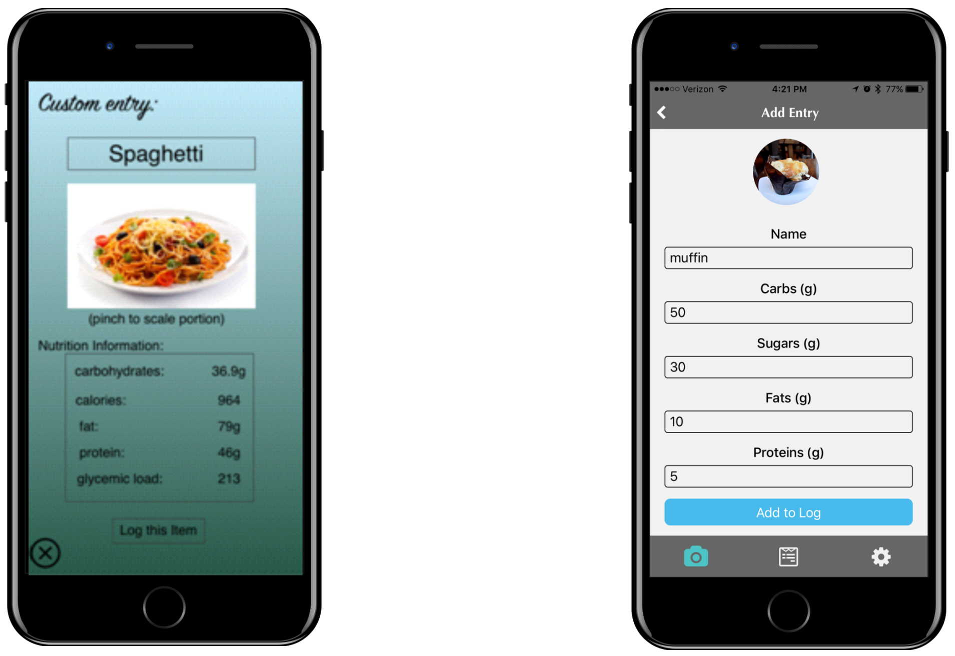


Figure 4.20: Medium and hi-fi home screens and navigation



Custom entry:

Spaghetti

(pinch to scale portion)

Nutrition Information:

carbohydrates:	36.9g
calories:	964
fat:	79g
protein:	46g
glycemic load:	213

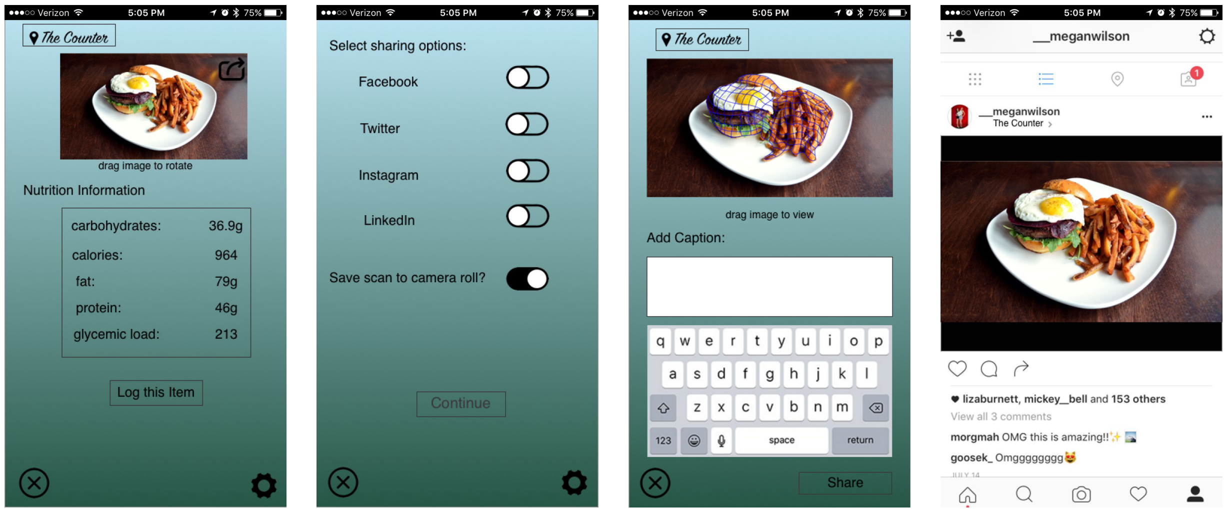
Add Entry

Name	muffin
Carbs (g)	50
Sugars (g)	30
Fats (g)	10
Proteins (g)	5

Figure 4.20: Medium and hi-fi home screens and navigation

The implementation of our share task changed drastically from our medium-fi prototype to our hi-fi prototype. Our heuristic evaluators made the astute observation that the sharing flow in the *Feast* medium-fi prototype differed greatly from the way sharing is implemented in most mobile apps. First, they identified that *Feast* reversed the conventional ordering of share screens in which the screen to create a caption appears before the user chooses which social media platforms to post to. They also noted that our medium-fi prototype did not contain any intermediary screen to warn users that they would be leaving the *Feast* app to view their post on the social media site they posted to. Additionally, there is no well-defined action for what the app would do if the user were to share a food item to several social media sites at once. How would *Feast* choose which social media post to show the user? Finally, navigating users away from *Feast* to another app without warning could pose security issues. To remedy these issues, we updated *Feast* to incorporate the industry standard of an intermediary screen that asks users for permission before directing them to view their post on a third-party social media site.[1]

[1] Due to development ability/technical constraints, our group was not able to incorporate this task in the high-fi prototype as of December 10, 2016. This feature is something we plan to implement in the future.

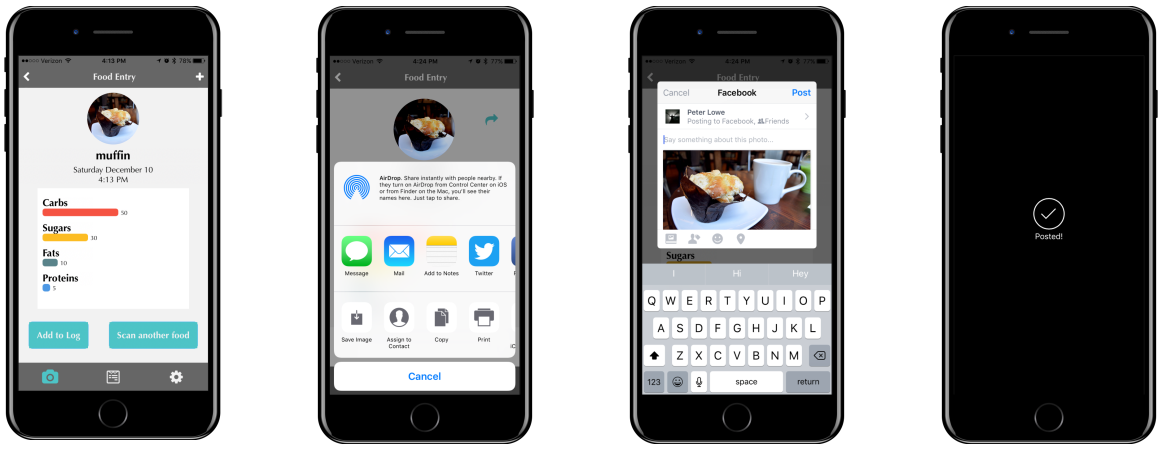


carbohydrates:	36.9g
calories:	964
fat:	79g
protein:	46g
glycemic load:	213

Facebook
Twitter
Instagram
LinkedIn
Save scan to camera roll?

♥ lizaburnett, mickey__bell and 153 others
View all 3 comments
morgmah OMG this is amazing!!✨
goosek_ Omgggggggg😻

Figure 4.21: Sharing in medium-fi prototype



AirDrop. Share instantly with people nearby. If they turn on AirDrop from Control Center on iOS or from Finder on the Mac, you'll see their names here. Just tap to share.

Figure 4.21: Sharing in hi-fi prototype

The final group of high-severity changes that our team focused on fixing from our medium-fi prototype to our high-fi prototype were largely related to consistent labeling and addressing flow mistakes. For example, we made sure to add name labels to all of the screens in which users view a food entry, allowed for users to delete items from their log to counteract the ability to add items, and ensuring that the language used in our labels was as informative and minimalist as possible. The images below provide some examples of small labeling and consistency changes that were able to remedy some of the more frustrating bugs in our medium-fi prototype.

Prototype Implementation:

We initially began developing an iOS app with the Xcode IDE (Integrated Development Environment) for our high-fi prototype before switching to React Native and the Exponent framework. React Native is a Javascript mobile UI builder library that allows developers to build native (running directly on the target platforms) mobile apps for iOS and Android. Exponent is a mobile app development and deployment tool that empowers developers to build and ship mobile apps for iOS and Android directly through the framework itself.

We decided to switch to React Native and Exponent for two main reasons. First, the development team was not very familiar with Xcode and Swift, the next-generation iOS programming language, and much more comfortable with web app development, in particular the Javascript programming language core to web dev that React Native is built on. Initial coding of the iOS app was thus a slow and constricted process, and we suspected we would not be able to build a powerful, well-designed, and compelling app given our current progress as of Thanksgiving break.

The second was that React Native, in particular as delivered through the Exponent framework, offered in our opinion a much smoother development experience that would enable us to build responsive, well-designed mobile apps for two major mobile platforms, an opportunity we thought justified the transition costs of learning React (the UI builder tool upon which React Native is built upon and inspired by), React Native, and the Exponent framework. React Native's declarative, functional-like design lead to cleaner and more readable code that made the entire development process--from building out the initial bare-bones version of the app to refining the functionality and design to debugging--a (relatively) pleasant and seamless process. While Swift's more low-level, configurable approach offers some advantages in terms of more fine-grained control over the functionality and design of an iOS app, we did not believe that those benefits applied in our case given our high-fi prototype, development, and time constraints. React Native's composed components design paradigm lead to more reusable, modular code that, in particular for that last sprint before the presentation/demo of our product and entire design process, helped us quickly and effectively improve our app across its various iterations. React Native's powerful Flexbox design algorithm, highly similar to how CSS manages design on the web, and capacity to control the presentation of the app given such features as the size and resolution of the screen (part of the new wave of responsive app development) allowed us very particular control over the app that lead to a professional-looking final product. The only significant issue we experienced with React Native and Exponent involved the relatively immature and limited libraries they offered, which made building some of the functionality of the app challenging and limited the incorporation of more advanced features. This constriction would really only become an issue if we decided to continue development of some version of this app, although React Native (and Exponent soon) allows developers to write native code in iOS and Android within a React Native app.

Given the ambitious nature of our app idea, we required some significant Wizard of Oz Techniques to build our final product. For the scanning process, we resorted to just taking advantage of the mobile platform's camera to produce the final "scan" of the food as there were no libraries involved for 3D scanning (even for iOS and certainly not for React Native) and trying to "simulate" a scanned photo through such techniques as 3D stitching did not seem necessary for the core use cases of our app. We also resorted to hard coding the classification of the scan in the food, associated nutrition facts, the blood sugar graph, the setting screen, and starter log entries. To fully implement the image classification and nutrition information extraction functionality, we would require beyond state-of-the-art deep learning techniques (even then this problem may be intractable given the ambiguity of an image and the diversity of food to start) and significant enterprise-scale database implementations. We had neither the expertise, time, funds, or network to implement those features. Regarding the blood sugar graph, although it is possible to connect a CGM (continuous glucose monitor) to someone that would output blood sugar readings via bluetooth or a wifi to a mobile app for presentation, we thought that implementing that functionality was also beyond the scope of our high-fi prototype. We hardcoded some initial log entries for the purpose of the demo of the log entries and the settings screen to give an idea for what information we would like the user to control the display of (we thought that implementing that functionality would complicate the display of the nutrition information far more than it was worth).

There are so many different directions we could go with this app. To start, we were interested in perhaps incorporating location-tracking so that, if users were at a restaurant, we could rely on available data to classify an image and output nutrition information. We would also be interested in working with medical researchers to more effectively determine what sort of data would be most useful for us to track and present to help diabetics. While we started a bit working with deep learning image classification libraries and testing them initially with some food scans (did not go too well), we could work a bit more on that end with deep learning specialists and determine in what direction we could go there. We could determine whether we could start with maybe detecting certain allergens, such as gluten, or just calories to provide information that many users in general would find very useful. We would also love to add a database so that log entries can persist beyond an app session.

Summary:

Feast is a diabetes management app that allows user to take a scan of their food, extract nutrition information, and log these entries to keep track of how their meals affect their blood sugar to make more intelligent decisions in controlling their health. Users can also share their scans with their friends through various communication and social media platforms. After realizing we were all interested in health, in particular chronic illnesses, we decided to focus on tackling diabetes management to address main pain points we identified in our needfinding.

A full cycle of implementing our ideas and following the design thinking and HCI methodologies was an intense and rewarding experience. Throughout the entire process, we especially enjoyed working with and learning from one another and receiving a lot of positive feedback on the problem we were addressing and the final app itself, and were humbled to receive the award for most innovative idea at the final presentation. Given how our idea had resonated so powerfully with so many people, we are considering extending this app further to help not only diabetics but others with special health needs to easily extract and manage nutrition information to improve their health and enjoy better health and lifestyle.