

Ola Diabetes Care

Ian Holmes, Tang Zhang, Albert Chen (hselin)

I. Problem and solution overview

Currently, diabetes patients typically receive treatment update only when they visit their doctor (once every 3 to 6 months). They also have to self-administer numerous measurements/treatments each day to maintain their health. Finally, the devices which patients use for their day-to-day health are expensive, hard to use and requires proprietary software to view/download historical/raw data.

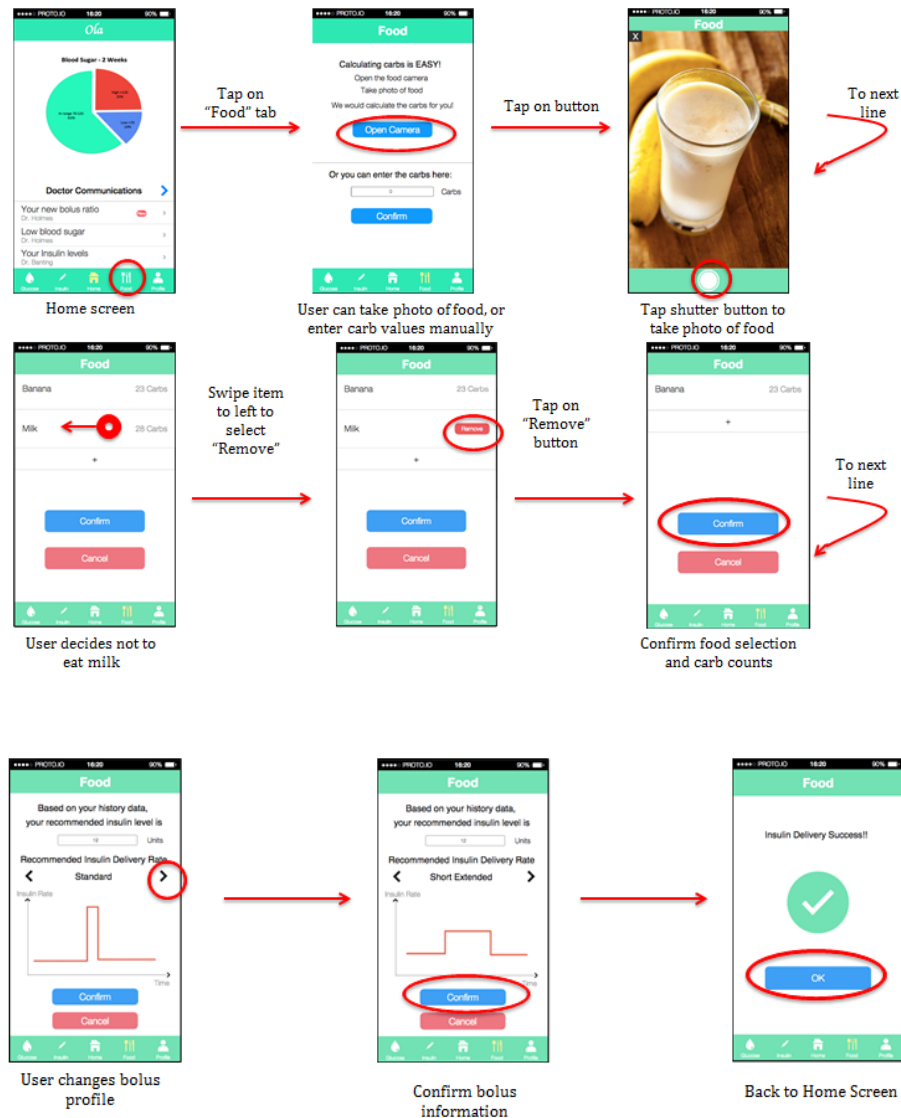
Ola is a system (smartphone application + wearable pod) that provides real time context aware care for diabetic patients through smart wearable devices and advanced data analytics.

Our mission is to enable diabetics to live a normal life without ever having to worry about their conditions.

II. Tasks

Our representative tasks are the follows:

- Eating a meal (simple)
 - Count amount of carbohydrates in food and determine the appropriate amount of insulin to inject.



- Emergency care (medium)
 - Emergency treatment and response when user experiences a severely low blood sugar event.

- Changing dosages (complex)
 - Help doctors determine and implement optimal insulin dosages.

III. Revised interface design

We revised the following interfaces to incorporate feedbacks from our medium-fi prototype.

I. Emergency screen (See Figures 1 and 2)

- Change: Add buttons to call emergency contact and medical help in emergency care information screen.
- Reason: Our low-fi interviewee thought it would help the bystander (caregiver) to have the ability to contact experts or emergency contact of the patient.

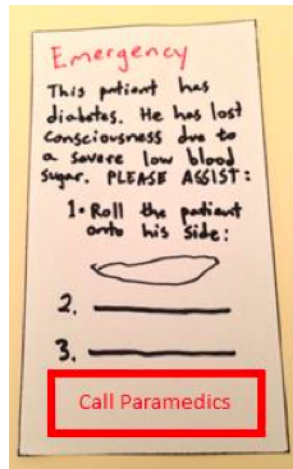


Figure 1 - Emergency screen feedback

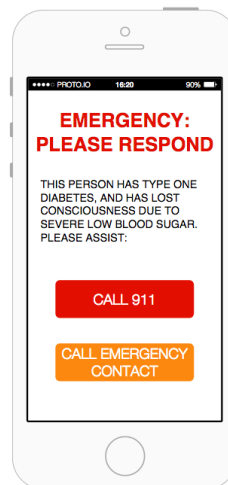


Figure 2 - Refined emergency screen UI

II. Food camera (See Figures 3 and 4)

- Change: Add instructions on the usage of food camera.
- Reason: Our low-fi interviewee was a little confused about the showing up of camera, and thought maybe she ought to take a selfie with it.



Figure 3 - Food camera feedback

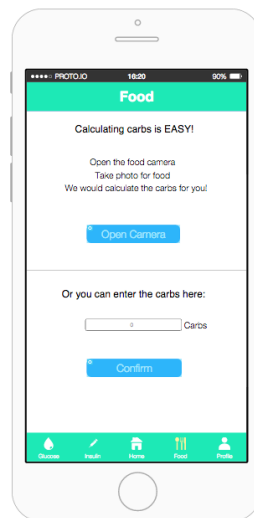


Figure 4 - Refined food camera UI

3. Insulin delivery screen (See Figures 5 and 6)

- Change: Provide built-in options for insulin delivery rate.
- Reason: Several of our low-fi interviewees thought a free-form input for insulin delivery rate is not essential and might lead to errors.

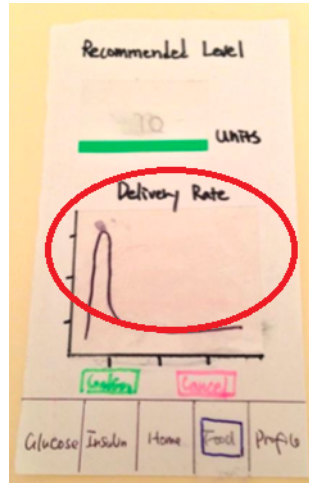


Figure 5 - Insulin delivery screen feedback

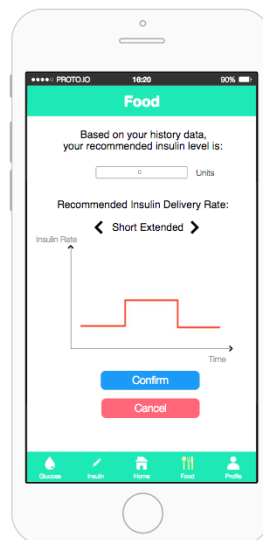


Figure 6 - Refined insulin delivery screen UI

IV. Prototype overview

We used proto.io to create our med-fi prototype.

We found the proto.io is good for:

- Implementing complex screen transitions/navigation.
- Simulating most interactions.
- Designing screen layout.

However, we found proto.io is not good for:

- Simulating the passage of time.
- Simulating an external event (e.g. receiving SMS text).

- Simulating camera.
- Generating dynamic content.
- Keeping lots of state.
- Concurrent editing.

Our prototype has the following limitations:

- Profile, Insulin and Glucose tabs are not yet implemented.
We felt they are not required for the completion of our representative tasks. Also, having these screens implemented may distract users from exploring the principal screens further. This also saves us effort and time on our med-fi implementation.
- Glucose/insulin history charts are not very detailed.
We felt having a low/medium fidelity mockup chart should provide enough affordances to the user, while leaving room for interpretation. This also saves us effort and time on our med-fi implementation.

Our prototype uses the following Wizard of Oz techniques:

- We pretended that the wearable pod for continuous glucose monitor and insulin/Glucagon injection is implemented and functional. It enables our user interface to provide the following functionalities.
 - Black out detection.
 - Glucose/insulin measurement and history.
 - Insulin/Glucagon injection.
- We also faked the food camera functionality as algorithm and code to implement object detection, identification and carbohydrate counting are too complex to implement and hard to get right.

Our prototype has the following hard-coded features:

- Historical insulin and glucose values on the home screen.
Justification: We don't have access to a real medical pod (see Wizard of Oz above).
- Food items in food camera screen are fixed.
Justification: There is no easy way to implement "real" camera functionalities with proto.io (see tool limitations and Wizard of Oz above).
- Doctor communiques are fixed and Dr. Holmes' email regarding "new bolus ratio" is always set as new unread email upon start of simulation.
Justification: There is no easy way to model external events such as receiving new email with proto.io (see tool limitations above).

V. Prototype screenshots

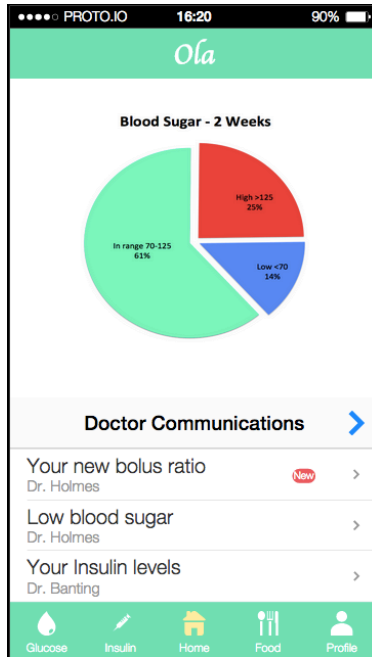


Figure 7 - Home screen

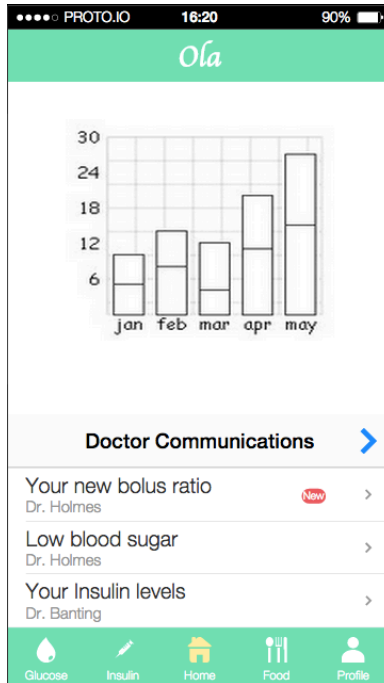


Figure 8 - Home screen

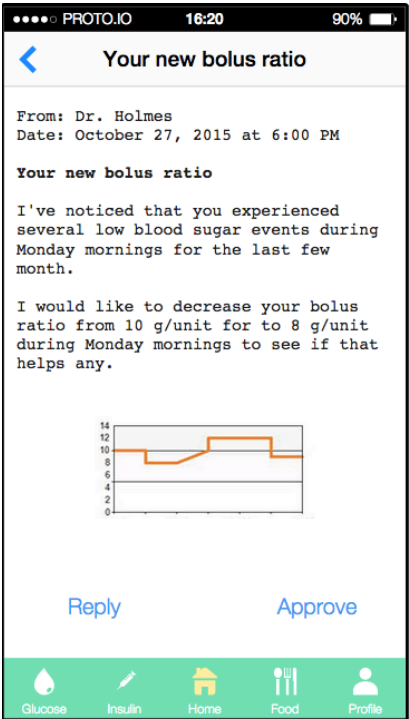


Figure 9 - Doctor communique

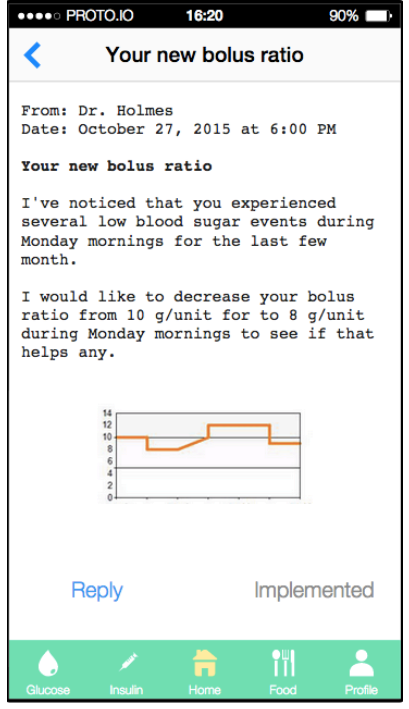


Figure 10 - Doctor communique

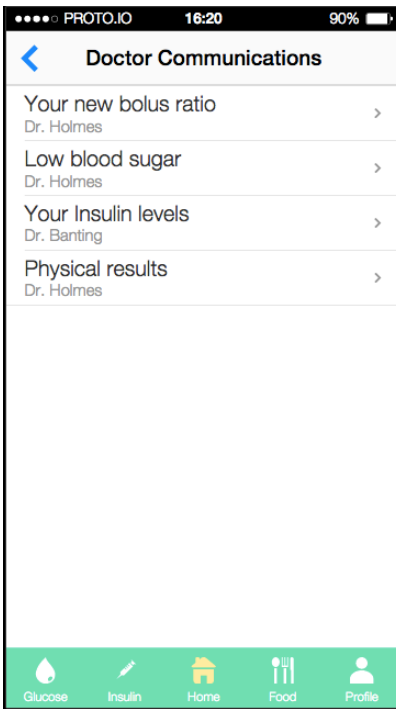


Figure 11 - List of doctor communications

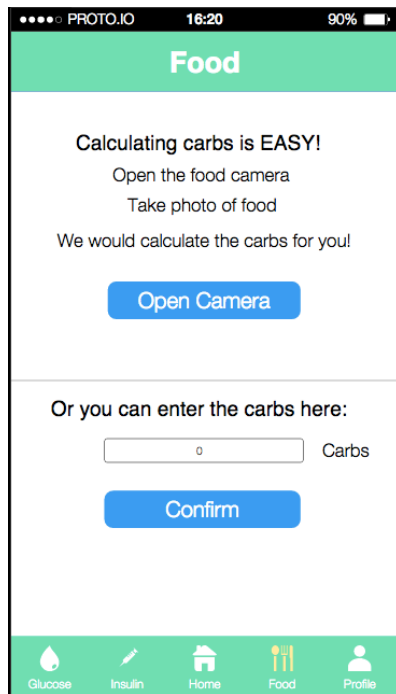


Figure 12 - Food carb count screen



Figure 13 - Food camera



Figure 14 - Food carb count screen

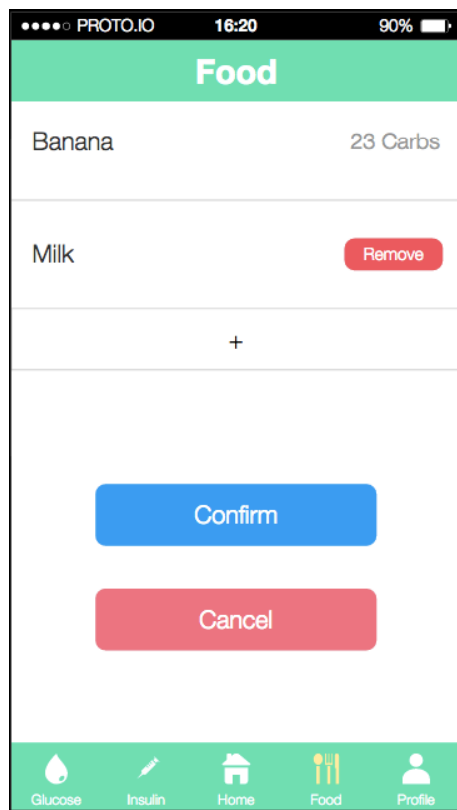


Figure 15 - Food carb count screen

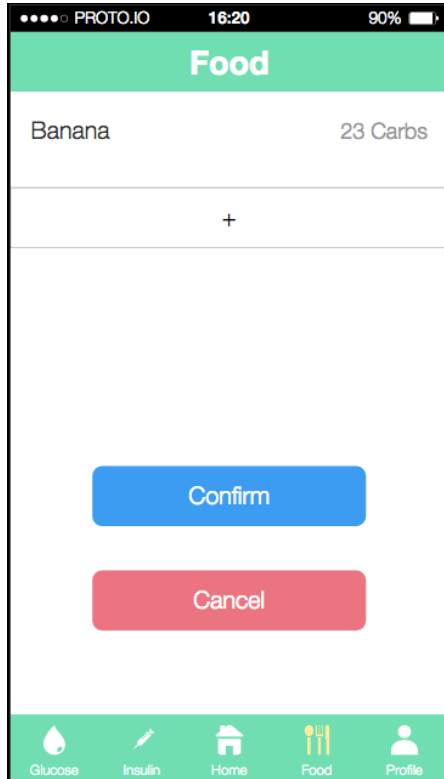


Figure 16 - Food carb count screen

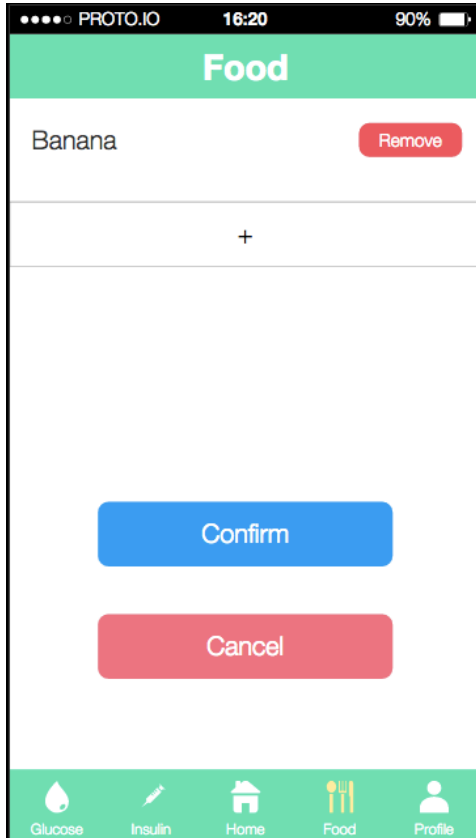


Figure 17 - Food carb count screen

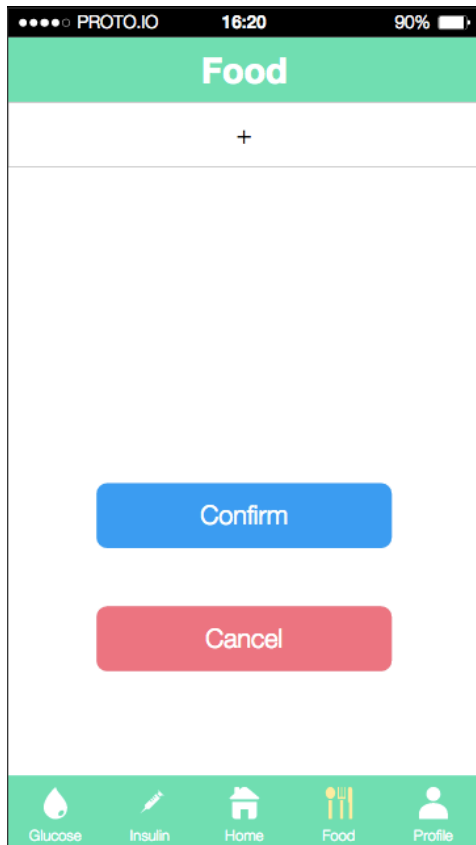


Figure 18 - Food carb count screen



Figure 19 - Food carb count screen

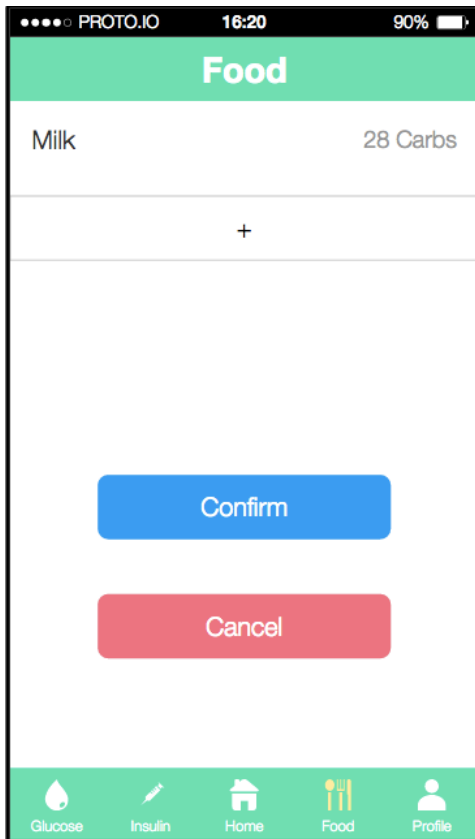


Figure 20 - Food carb count screen

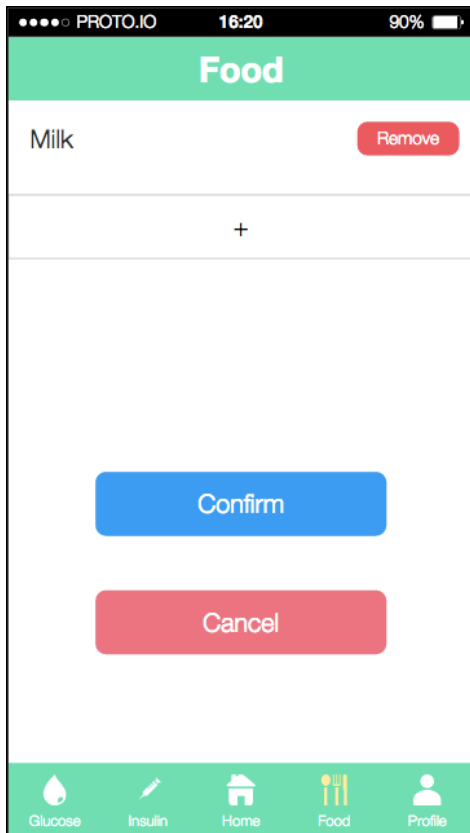


Figure 21 - Food carb count screen

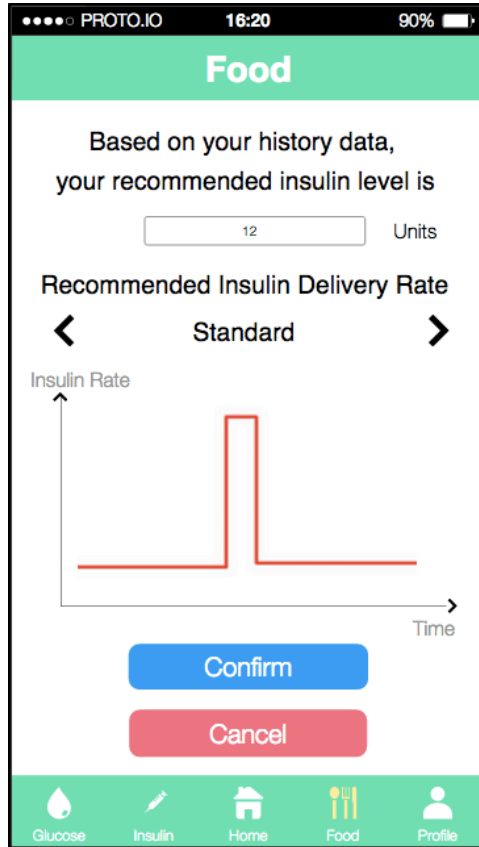


Figure 22 - Insulin delivery screen

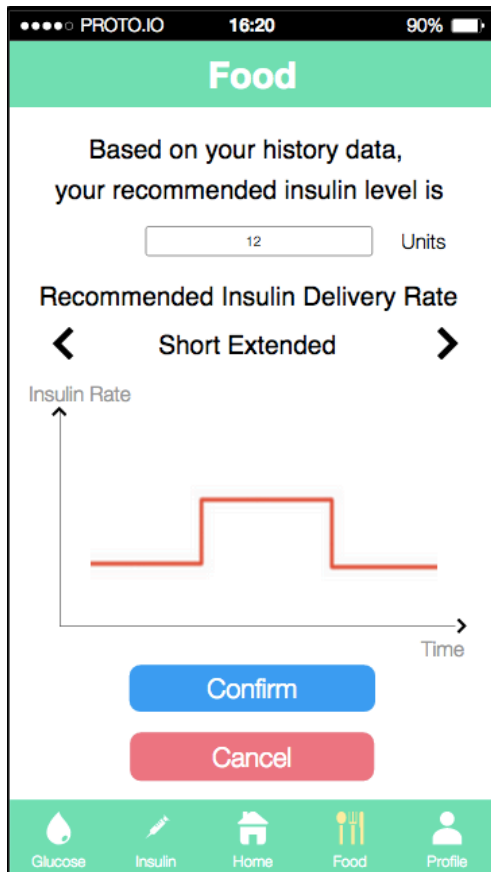


Figure 23 - Insulin delivery screen

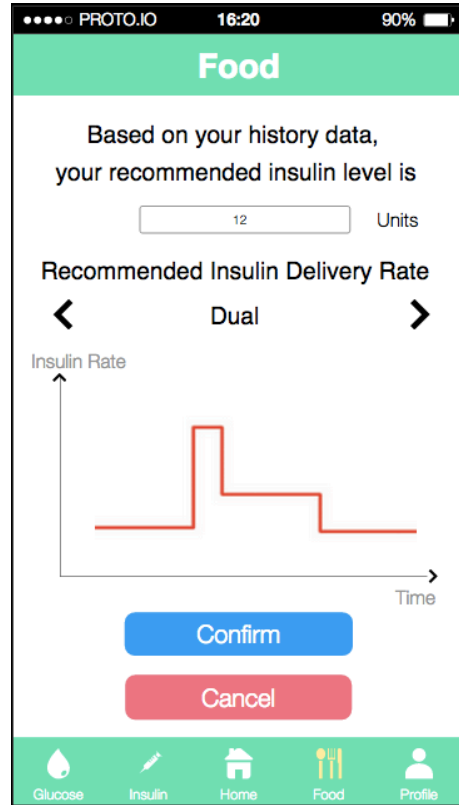


Figure 24 - Insulin delivery screen

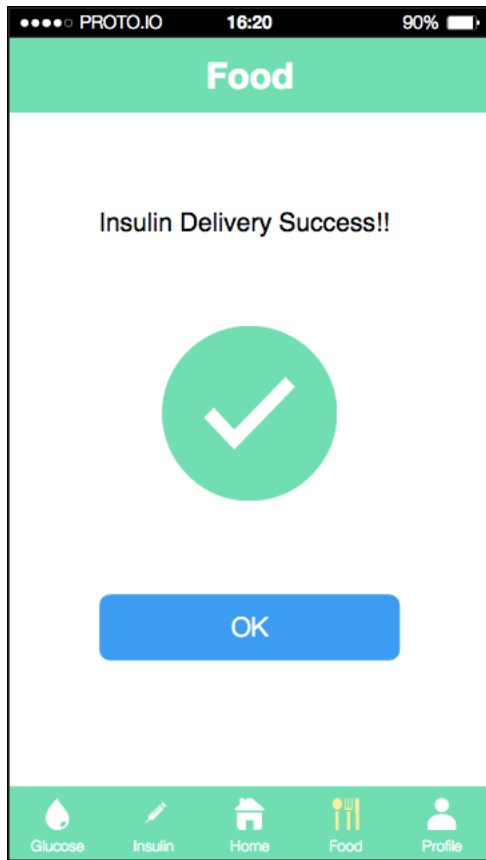


Figure 25 - Insulin delivery confirmation screen

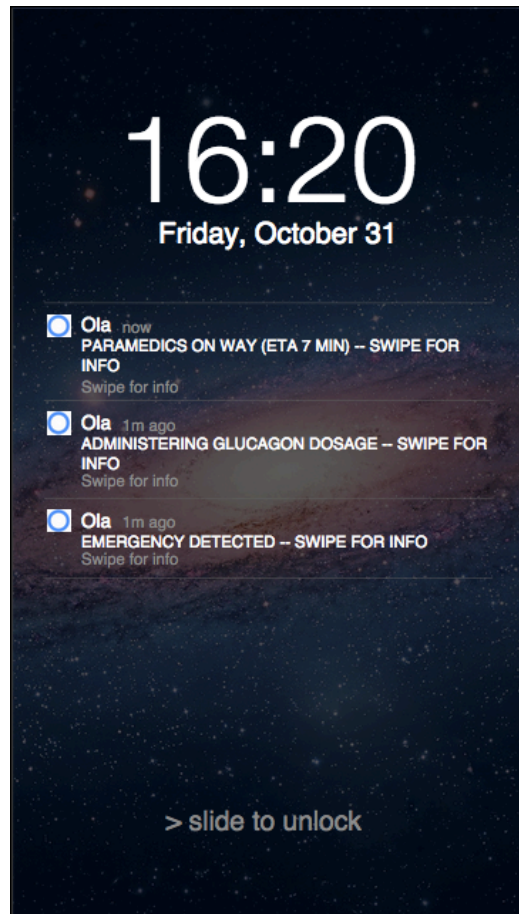


Figure 26 - Emergency detected screen

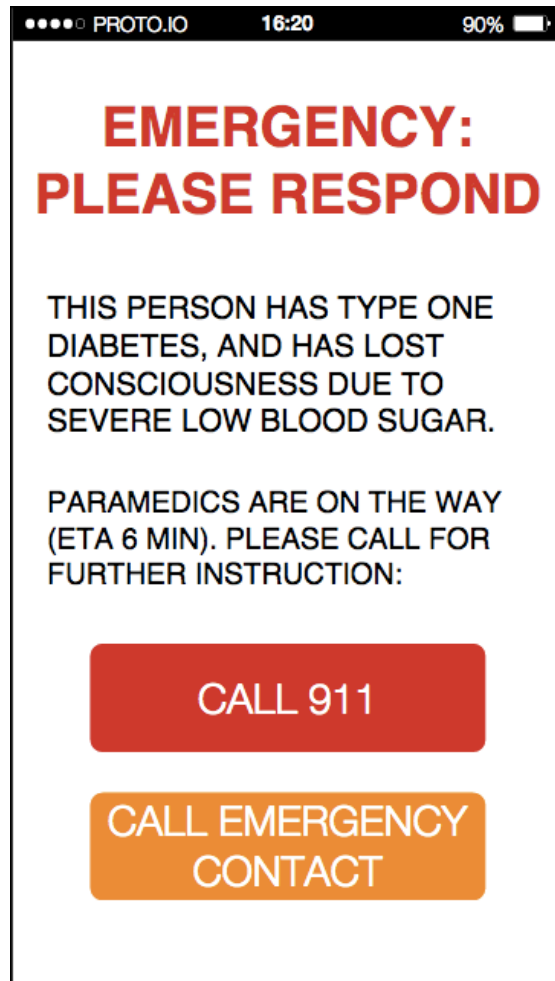


Figure 27 - Emergency care information screen



Figure 28 - Coming soon screen

Prototype URL <https://iholmes.proto.io/share/?id=005af9f9-abd3-44f4-969e-0fff49d6112c&v=1>

README URL http://stanford.edu/~hselin/cs147_assignment9_readme.pdf