So why iREACH?

Problem:
In the United States of America, 29.1 million people have diabetes. This is about 1 out of every 11 people. Also, 86 million people have prediabetes. Without weight loss and moderate physical activity, 15-30% of people with prediabetes will develop type 2 diabetes within 5 years. Currently, at least 1 out of every 3 people will develop type 2 diabetes in their lifetime.

- A total of $245 billion is spent on medical costs, lost work, and wages for people diagnosed with diabetes.
- Adults with diabetes have a 50% higher likelihood of dying than adults without diabetes.
- Medical costs for people with diabetes is twice as high as for people without diabetes.

Solution:
- You can prevent or delay type 2 diabetes by: losing weight, eating healthy, and being more active.
- You can manage diabetes by: working with a health professional, eating healthy, and staying active.

So why iREACH?
- iREACH brings the next level of diabetes prevention by allowing for closer doctor-patient relationships and interaction.
- iREACH streamlines patient food and activity information straight to their personal physician.

iREACH - intelligent RemotE diAbetes Coach

Sponsor: EPSL Embedded Pervasive Sensors Lab
Team: Jeremy Martinez, Kyle Avrey, Jacob St. Hilaire, James Jessen, Faustino Lukolo
Mentors: Hassan Ghasemzadeh, Ramyar Saeedi, Ali Rokni

Why iREACH?

Architecture

The architecture of our entire system spans multiple machines. The API and Website are hosted on servers with available access from an internet connection. The Database is offline and on the same machine as the API because the API interfaces with the database so often. We use a RESTful approach here which means that the API is stateless. It is TLS encrypted and most API calls are protected through authentication. The client-side applications run in the browser and Android application environments. The browser application was mainly designed for doctors and administrators usage, although there are views for a patient to use as well. However, the patient will mostly only use the mobile application.

Database

The database is done using a NoSQL approach MongoDB. The structure of our data is relatively flat, meaning there is not a demand for having to join tables too often, so NoSQL was the perfect solution. The data spans three stakeholders: patients, doctors, and administrators. For diet information, we reference a public API maintained by the USDA.

Future Work

The iREACH project has many layers to it and is in great shape for expansion. In order to fully conceptualize this project the following features can be further developed and refined:
- Feature Extractor
- Doctor tokens
- Data Mapping

Also, the API and website can always be expanded upon to include new features like doctor/patient communication, different data visualization, more diverse diet constraints, etc.

Glossary

- Accelerometer: an instrument for measuring acceleration.
- Gyroscope: are devices that sense angular velocity.
- Magnetometer: are measurement instruments used for two general purposes: to measure the magnetization of a magnetic material like a ferromagnet, or to measure the strength and, in some cases, the direction of the magnetic field at a point in space.
- Feature: is an individual measurable property of a phenomenon being observed.
- Algorithm: is a procedure or formula for solving a problem.
- Synergy: the interaction of elements that when combined produce a total effect that is greater than the sum of the individual elements, contributions, etc.

Mobile Application

The mobile application is comprised of a User Interface and an Engine.

Diabetes affects people of all ages, so it is very important that our application is easy to navigate and fast to master.

The concept of easy to use, fast to master can be seen in how the engine takes care of all the users needs behind the scenes without their input. The sequence diagram to the right shows how the app flows depending on the user wants.

For example, say the user wants to log their meal. Clicking the “Log Food Data” button, the app changes the layout to the “Food Page” layout. After filling out the food name and serving size the user then clicks the “Save” button which causes the data to record the current date, time, user, food ID, and serving size. All of the gathered required information is then sent to the awaiting REST API for processing.

This Android Application accomplishes the task of providing an intuitive user interface accompanied with the full power from the online REST API.

Acknowledgements

- The team would like to thank the subjects used for data collection.
- Sepideh Mazrouee and the EECS department

Team: Achievers