

Bio-Behavioral System to Motivate and Enforce Heart Health

Jake Farrar, Austin Pederson & Jacob Lemon

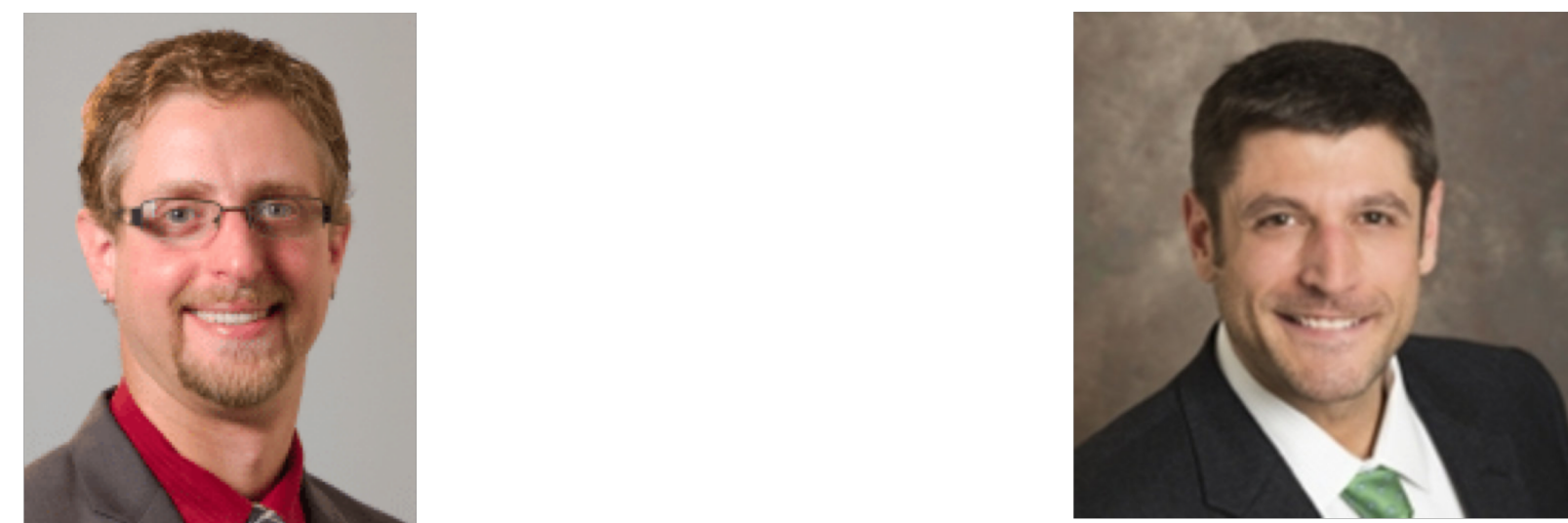
CEIAS Computer Science Capstone, Northern Arizona University, Flagstaff,



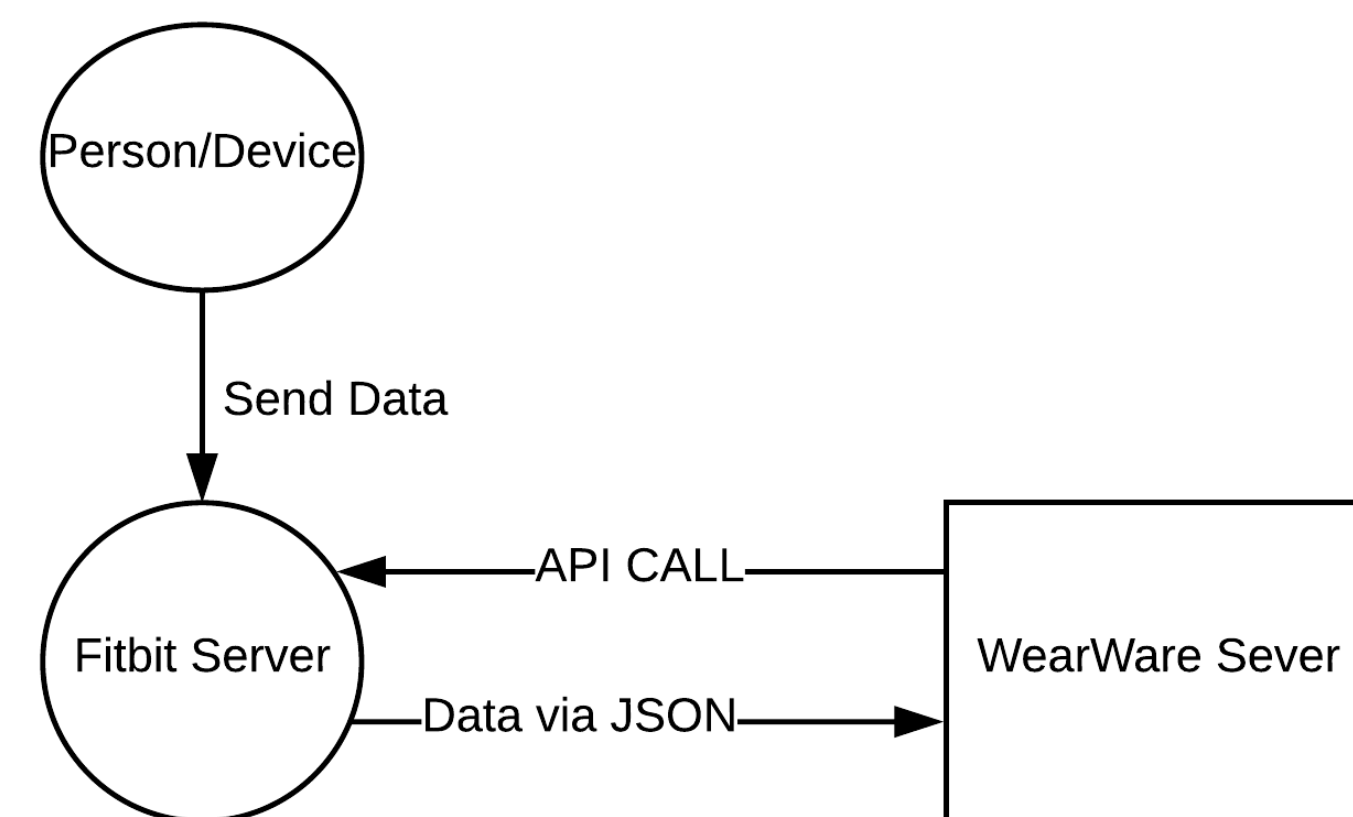
Office of Undergraduate Research and Creative Activity

Motivation & Problems

WearWare is an already existing Django program with a web application that are clients Dr. Kyle Winfree and Dr. Gregory Dominick are using to help facilitate research in regards to heart health through the use of Fitbits as heart related diseases are quite common.

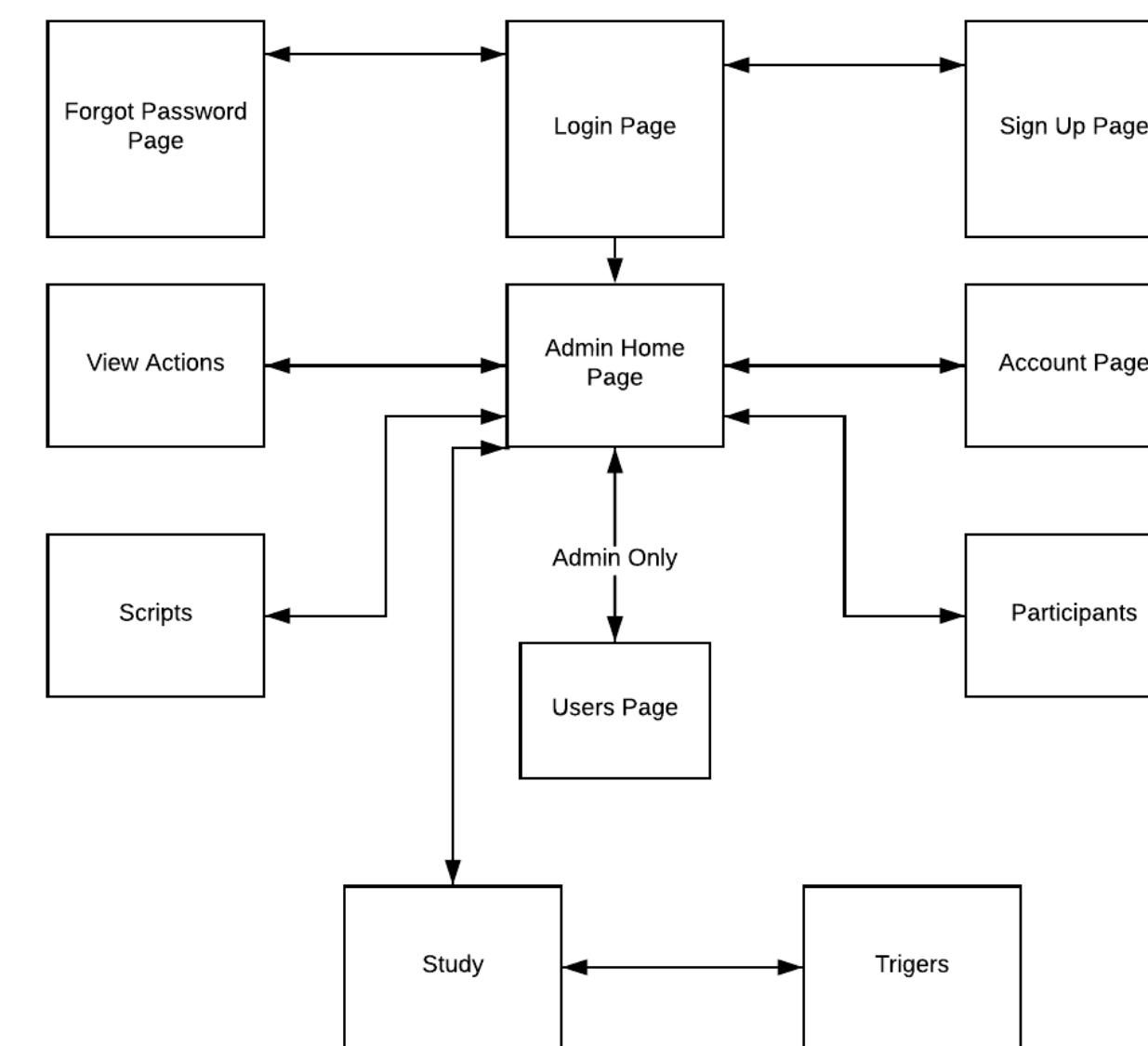


Our sponsors, Dr. Winfree and Dr. Dominick



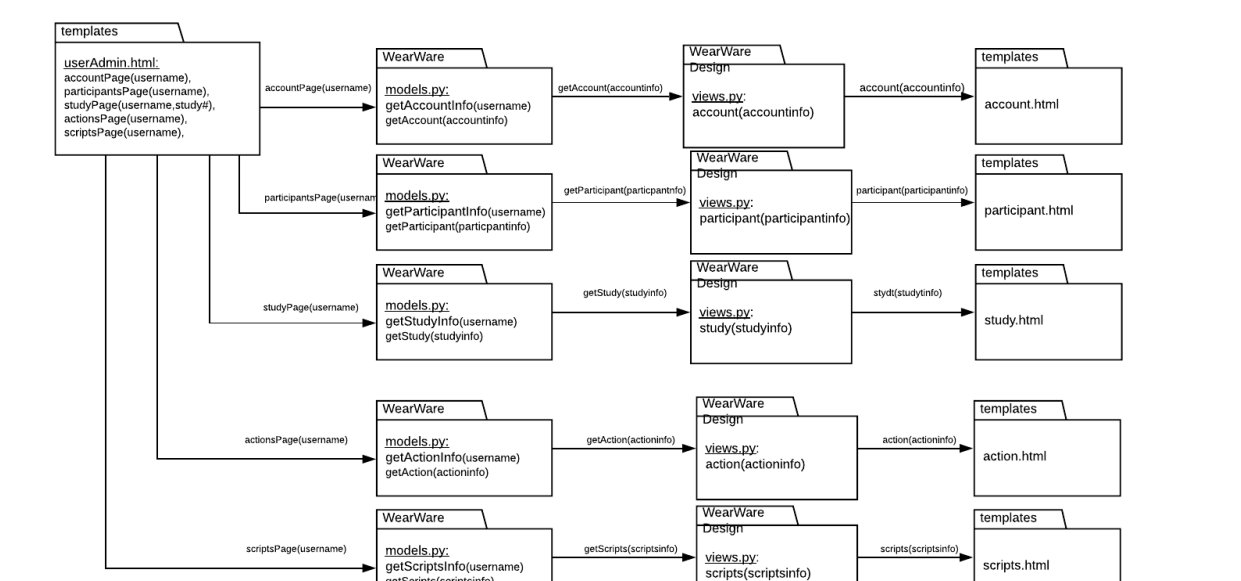
Their version of WearWare that calls the Fitbit Server to grab the data for specific users. It works but has problems. These problems include localized system, poor user interface, no contact tool, and limited analysis.

Solutions



The team's solution for the updated UI. Each box represents a page and each arrow indicated that the page can be reached from the previous page.

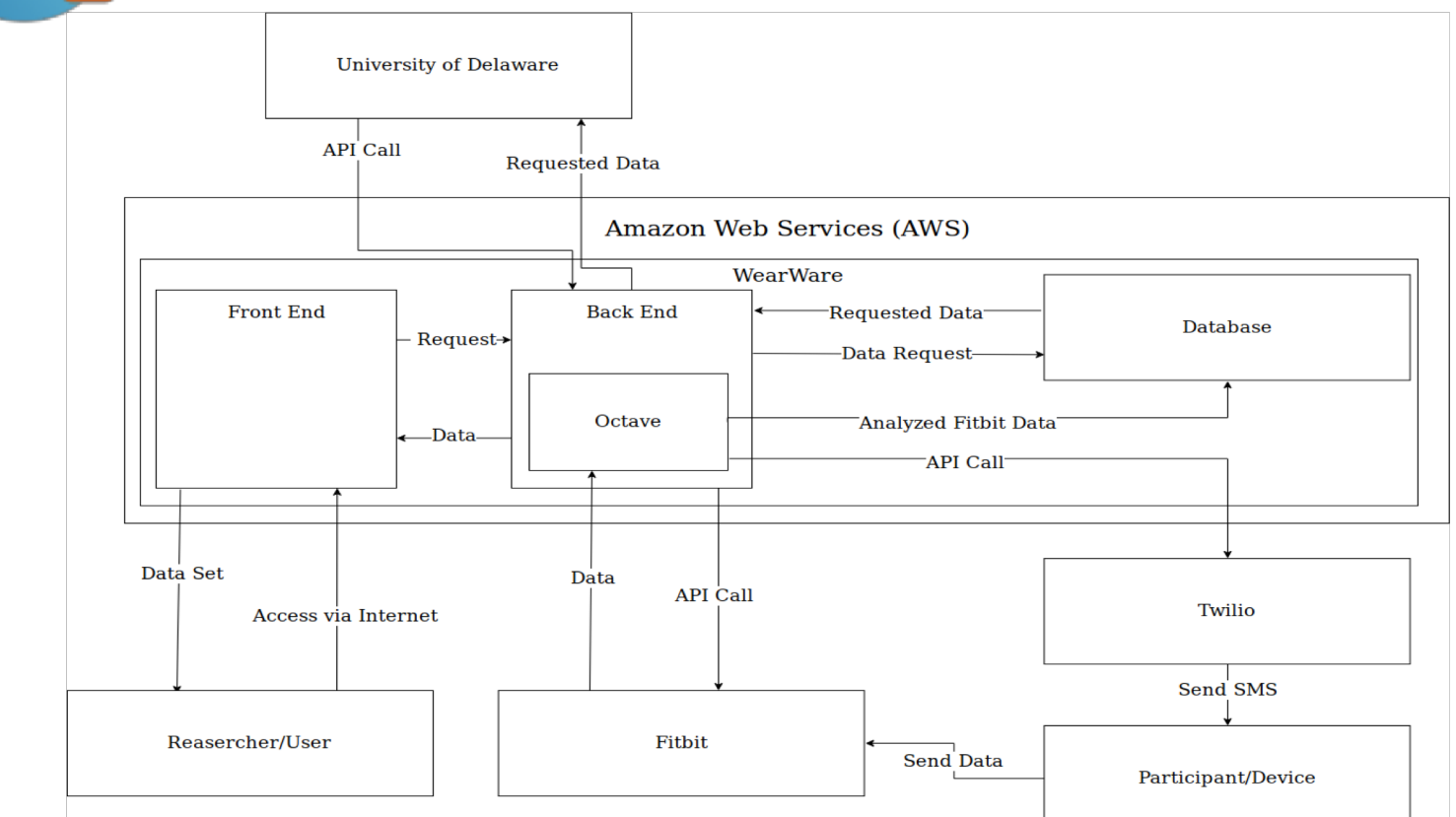
The team planned to solve the problems via API access, updating web app, utilizing SMS, and Octave support.



A flow chart for a page would run in Django

Implementation Overview

- Amazon Web Services: House the WearWare Application
- Django Web Framework: Base code for WearWare
- PostgreSQL: Storing all data from Users/Participants
- Octave: Language to analyze data
- Twilio: API to send out SMS

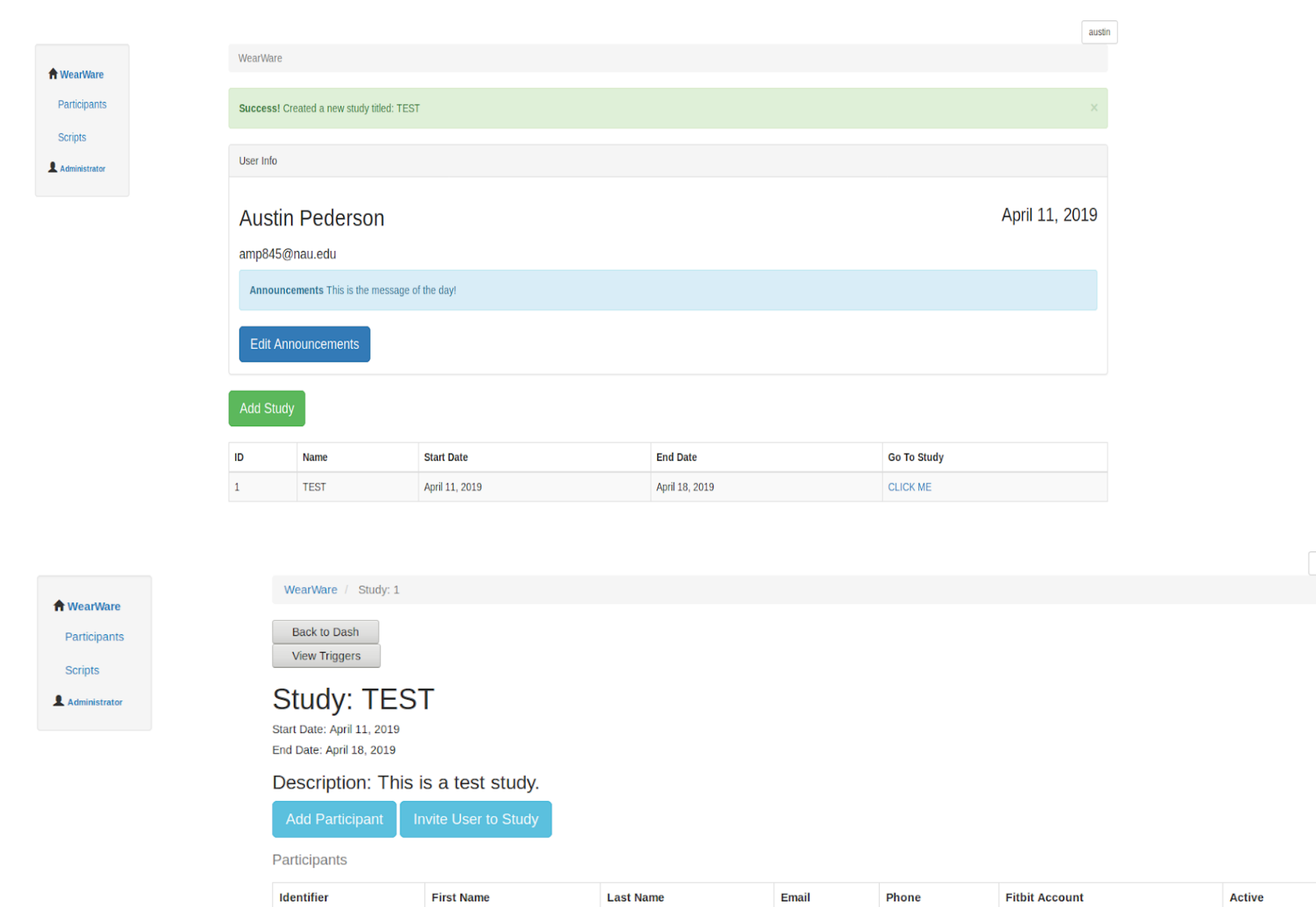


The new system overview of WearWare with the team's solutions.

Challenges

The team encountered a few challenges that included implementing a API, working in a new language and a struggling team dynamic. The team over came these challenges by working with our mentor to discuss the problems as well as working with our sponsors about exactly what they wanted from the software.

Final Product



These images show the completed web application of WearWare. They showcase the main dashboard as well as a sample study. They focus on solving the problem of the poor UI.

Testing

The team completed many tests to ensure the correct results were being shown and the program could be easily navigated. Unit testing was used to make sure that the code was running correctly and producing the correct results. Integration testing was used to check functionality of key features. The team worked with Dr. Winfree for making the design as usability testing. Django's built in features were also used for testing.

Key Features

The team addition to WearWare adds many new features to the existing program. WearWare now has texting functionality to talk to participants via texts. The system also allows for multiple researchers to work together on studies which was not possible before. The web application is more user friendly. WearWare is still collecting data and processing it for the user.

Future Work

The team hopes that WearWare will be using to help facilitate research and help many live a heart healthy lifestyle. The program should be able to be improved in many ways as the sponsors sees fit.