Tiny House Project
10/9/15

Project Owner: Joseph Ou
Initial Scrum Master: Shivam Dave
Team Member: Sargis Yonan
Team Member: Reza Barghi
Team Member: Sean Ashe

Project Overview

- The Tiny House Competition is a statewide competition hosted by SMUD, the Sacramento Municipal Utility District
- Universities around California compete to make the most efficient tiny houses (under 400 sq ft)
- UCSC, partnering with Cabrillo college, will be working on a solar powered and “smart” tiny house project that will be implemented in the Watsonville area
  - Property built in Watsonville area provides affordable housing in exchange for data collection
  - Additional plans to design Tiny House to be modular in order to integrate groups of tiny houses into row houses
- Goal of the team is to create a modular sensor network for smart appliances, collect usage and environmental data, and analyze the data collected

Sprint 1 Overview - Research

User Story 1 “As a product owner, I want to establish the scope of our project to have a complete view of the hardware and software required for the implementation

- task 1: Hold discussion with project leader at UCSC and IDEASS team and brainstorm realistic requirements of what metrics we want to detect.
- task 2: Brainstorm with the team and figure out the hardware required to perform measurements based on the metrics established.

User Story 2: “As a developer I want to research a basic web application skeleton.”

- task 1: Understand existing dashboard developed from previous research done at UCSC
- task 2: Understand programming languages used, and use prior dashboard to create basic mockup & version of web app

User Story 3: “As a developer I want to understand data being handled and software used to model data.

- task 1: Learn to access the data such as handling the objects and how they may be properly used and implemented into the front end interface

User Story 4: “As a developer I want to understand the existing hardware being used.”

- task 1: Become familiar with sensor technologies used and current infrastructure already implemented by previous researchers

Sprint 2 Overview: Implementation

User story 1: “As a developer, I want to integrate sensors into a network so I can collect their data.”

- task 1: Implement code that will allow the central microcontroller to receive information from multiple sensors
- task 2: Implement code that will push data from the microcontroller to a storage unit and the database

User story 2: “As a developer, I want to create basic web dashboard that handles either incoming or sample data.”

- task: Implement researched programming on web application, use user input to display sample data in manner that would be required, show sample models created on web app

User story 3: “As a developer, I want to parse data in user friendly manner that can be received from the database and handled by the front end web app”

- task 1: Create sample models that either handle real/test data, confirm model types with users

Sprint 3 Overview: Live Data Handling and Testing

User story 1: “As a user, I want to be able to receive information relevant to me and my lifestyle.”

- task 1: implement algorithms to analyze data in the database and provide recommendations
- task 2: implement the display of live data on the dashboard use results from analysis to display a suggestion

User story 2: “As a developer, I want to complete models in a manner relevant to our project scope that handle incoming data and create recommendations for the user based on incoming information.”

- task 1: Test sample models
- task 2: Handle large amounts of data effectively from the database
- task 3: Showcase data cleanly in models

User story 3: “As a product owner, I want to conclude my part of the project and facilitate further research on the project.”

- task: Prepare for further implementation in front end, and confirm/test with user base

Our Task

Backend Development: Hardware - Sargis Yonan & Joseph Ou

- We will integrate a frequency response mechanism to more efficiently use power from the grid
- We will take information from frequency meters and temperature sensors, and control various appliances (water heaters, washer/dryers, refrigerators) to operate or shut down when optimal
- We are using embedded ARM microcontrollers to control the system via memory-mapping I/O ports and pins through an OS (Linux (Debian))

Our Task

Backend Development: Database - Sean Ashe

- **Goal:** Establish an internal database system for consistent data collection
  - Appliance usage trends
  - Resource flows (water, energy usage, temperature)
- **Motivation:** Provide a foundation for improving the intelligence and dynamism of Tiny House

Our Task

Frontend Development: Data visualization - Reza Barghi

- Create models and graphs of immediate data that may be implemented into a user friendly application.
- Relay between the database and the front end dashboard; parsing the data for the dashboard.
- End Goal: Create a model of the power usage of multiple households to implement better techniques for energy consumption.
- Ex: Everyone usually works from 8am-5pm so in most households energy consumption should be at a minimum during those times.

Our Task

Frontend Development: GUI, Data Visualization - Shivam Dave

- Display incoming sensor and power information through a flexible and easy to navigate web application
- Create a responsive design through media queries and user agent parsing to create a consistent user experience across a range of devices
- Allow users to understand, visualize, and compare incoming and collected data
- Provide GUI for clear and concise status reports that recommend responses to shifts in frequency and power consumption

Challenges/Risks

- Broad scope
  - need to coordinate with non-CS teams
  - need to reduce the scope of the project to a manageable size and focus on deliverables
- Need to decide on what implementations and techniques work
- Issue of privacy
  - real people are living in the Watsonville homes; data collected must be kept private

Technologies

Programming Languages: C, Cython, Python, ARM assembly

Web Framework: Javascript, HTML, CSS

Equipment: ARM microcontroller with a Linux Kernel, Atmel Atmega AVRs, an FPGA, ADCs, DACs, band-pass frequency filters, AC voltage frequency meter, water flow meter, thermistors, wireless emitters and receivers (IEEE 802.15.4), circuit board printing, high-powered switches, and more sensors, actuators, and components.