# Envismo: A Mobile Platform for Collecting and Visualizing Energy Usage Data


Anuj Tewari
University of California, Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720-1776
anuj@eecs.berkeley.edu

Wesley Willett
University of California, Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720-1776
willettw@eecs.berkeley.edu

Nanheng Wu
University of California, Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720-1776
nanhengwu@berkeley.edu



**Figure 1. A Nokia N810 Internet tablet displaying the Envismo mobile interface for mapping energy usage data. An attached Arduino microcontroller and environmental sensors provide supplemental environmental data.**


## ABSTRACT

We have constructed Envismo (**en**ergy-**vis**ualization-**mo**bility), a prototype of a real-time energy-monitoring platform that pairs common mobile Internet devices (Nokia N810 internet tablets) with cheap, readily available environmental sensors and a web-based datastore tied to an existing energy-monitoring network. This allows for the collection of real-time fine-grained environmental and energy usage data that can supplement and contextualize existing measures like electricity consumption and water use. Users can interact with visualizations of this data on the mobile device in order to evaluate energy consumption patterns, diagnose electrical problems and inefficiencies, and explore their own energy use. A current prototype uses utilities data from the UC Berkeley Green Building Research Center (http://greenbuildings.berkeley.edu), and allows us to display current and historical data from energy, steam, and water meters across the Berkeley campus and supplement it with temperature and luminosity information.


## 1. INTRODUCTION

The monitoring of energy usage at personal, institutional, and community scales is becoming an increasingly important topic as we struggle to cope simultaneously with increased resource scarcity and the environmental consequences of increased consumption. However, in order to alter energy usage patterns, diagnose and treat inefficiencies, and raise awareness of the individual and collective costs of our energy consumption, accurate contextualized measures of energy use must be available. This is, in part, a problem of data collection and delivery. In most cases data about energy usage is exposed to users only at a very abstract level, if at all. Most energy consumers see, at best, one aggregated meter reading per month in the form of an water or electricity bill showing total use for an entire unit. Measurements of such coarse granularity do not allow a consumer to readily gauge the impact of individual devices, spaces, or practices on energy use. Without targeted measures of energy consumption a building inhabitant may not, for example, be able to discern whether or not his or her equipment or lighting contributes to the total energy use of a building. Likewise, building maintenance people may have a more difficult time discovering and taking steps to remedy inefficiencies.

Even when real-time, targeted data on energy usage is present, it can become much more meaningful when paired with additional information about of environmental conditions and the context in which spaces and devices are used. For example, measures of temperature and illuminance can help users understand when heating and lighting in a space is being used unnecessarily. Similarly, measures of activity in a space can be used to help determine whether or not activity and energy use are related. One might ask: Are lights left on when people leave a building for the night? More targeted measurement devices like EMF sensors can help identify devices that are large consumers of power. In general, a better understanding of energy usage requires finer-granularity sampling (in both space and time) of energy use and supplemental, preferably without adding additional infrastructure.

In that light, we wish to provide users with access to detailed and up-to-date energy usage information for the spaces they use. More importantly, however, we wish to allow them to take part in the collection and utilization of this data. We hypothesize that access to environmental information in real time, on location, and in tandem with additional contextual information allows energy users to make more conscious and effective decisions about their energy use. We also believe that allowing active participation in the collaborative collection and analysis of this environmental data engenders a greater investment in the process and stands a greater chance of engaging users and encouraging them to make real changes in their energy usage.

To this end, we have constructed at a prototype of a real-time energy-monitoring platform by pairing common mobile Internet devices (Nokia n810 Internet Tablets) with cheap, readily available environmental sensors and a web-based datastore. Our project is driven by the observation that mobile-devices enable researchers to collect more granular data and comments not captured by the current system and contribute to the database. Moreover, since most mobile-devices today are GPS enabled an interested user is able to monitor his/her energy consumption through out the day.

Such a system could be useful for monitoring use on the scale of a single home, an office, complex, or even a city. In each case, it is important to have access to some basic energy usage information, most likely at or near-real time electricity usage. Ideally, such a system would also provide an existing repository of usage data on which to bootstrap the system. We tested our prototype using data from the Berkeley Campus Dashboard project, which currently publishes data on electricity, steam, and water usage for a number of buildings on the UC Berkeley campus.

# 2. Motivation / Usage Scenarios

We focused our initial design discussions around three use scenarios. While the system as it stands does not directly implement all of the functionality described (nor are the situations necessarily realistic), these personas and scenarios illustrate our vision for the system and help illustrate the kinds of mobile interactions and use cases we would like to support.

## 2.1 Scenario A:

Jason is a researcher from the Berkeley Institute of the Environment. He is working on a project that collects supplemental data related to energy use and study their correlation. Envismo's data collection module allows him to plug existing sensors including temperature, a light, and EMF meters into the system and store data from them at a permanent web-accessible location. Jason is also interested in logging events and incidents that are energy related to supplement his sensor-based data. Envismo allows him to do this by adding text annotations to recorded data. One day while he is in the Hearst Mining building attending an event Jason notices an excess of active lighting and electronic. Feeling this information may be relevant later, Jason uses his mobile device to collect additional sensor information adds a comment on the data he is collecting.

## 2.2 Scenario B:

Jenny is a student at UC Berkeley and she is an environmentally conscious person. She is particularly interested in the energy use situation on campus and would like to learn more about it. Jenny is interested in observing resource usage trend ssuch as average use on per-building bases, or building-to-building comparison. However, she is even more intrigued by the "story" behind all the data. For instance, when Jenny discovers a spike in electricity use in the Life Sciences Addition building she wants know what could have accounted for such event. Envismo helps her with both tasks. When Jenny is on campus she can specify an area view nearby energy readings. The system allows her to establish the area by choosing a location as the center and selecting how large she would like the area to be. Because the system is location-aware Jenny can quickly check usage data and environmental data from her current location. Envismo offers many views of the data to Jenny such as daily to annually average that may reveal abnormalities that she did not previously know about.

During her studying of the daily average data of the Hearst Mining Building, she discerns a period of about three hours in the evening when the consumption of electricity is higher than its average level of during that time. She opens the comment view on the data and finds several annotations made by other users. Based on other users' comments she realizes that an event was held in the building during that period and that a lot of energy was used to power the projector, lights and other appliances.

## 2.3 Scenario C:

Joe is a manager in campus facility management. It is Joe's goal to keep resource use as efficient as possible. The current infrastructure allows Joe to be notified if there are incidents regarding equipment conditions on campus such as a water pipe leakage or electricity overload. However, to detect the inefficiency of resource use more contextual data are needed to aid Joe's analysis. Because Joe spends most of his day moving around the campus, he uses his mobile device, rather than a desktop PC to monitor environmental and energy data. Through Envismo, Joe can access visualizations of real-time readings from different buildings and compare their current performance to historical averages. These visualizations enable Joe to discover unusual performance and quickly remedy problems.

Sensor data collected by other users helps Joe to better understand the building-level data. One day, Joe observes a high level of steam use in Cory Hall. He toggles to the visualization mode and selects the temperature data in Cory Hall. A plot of user-collected temperature data suggests that Cory is much hotter than average. Meanwhile, a motion sensor plot reveals that the building is currently only lightly occupied. Using this information Joe concludes that it may a good idea to reduce overall steam throughput to Cory Hall to conserve energy.

# 3. RELATED WORK

In recent researches much effort has been made to raise the public's awareness of resource consumption. Several universities have created projects to gather energy usage information on campus and build visualization of the data such as the Campus Power project by the Live Building Lab of Queen's University [18] and Energy Use by MIT [12]. In this space, Envismo is closely related to the ongoing work of the Berkeley Campus Dashboard project [23]. The Berkeley Campus Dashboard project is an open, web-based platform for visualizing and annotating resource consumption data on the UC Berkeley campus. The Berkeley Campus Dashboard has a repository of historical data of consumption in electricity, steam and water from over 100 buildings; it also continuously collects such data from measurement meters installed at approximately ten buildings.

Our project is different from the Berkeley Campus Dashboard in that we focus primarily on providing a mobile energy-monitoring platform with location awareness. Because data in Envismo is coupled with its GPS coordinates it allows users to visualize data specific to the user's current location or any location selected through the user interface. Moreover, our system allows users to collect additional environmental data to supplement readings available from utility meters.

Using mobile devices as environmental sensors is another active area of research. The Harbor Community Studies team as part of the UCLA Urban Sensing project monitors PM2.5 exposure level in impacted neighborhoods. The CarTel project in MIT builds a distributed network using cars and smartphones to collect sensor data [16] The Common Sense team is developing mobile environmental sensing platforms to support community action and citizen science [22]. We use Arduino[1] microcontrollers, which can able to collect reading from a variety of economical sensors such as light, temperature, and noise sensors. The sensor platform allows users to add supplementary data that are not captured by meters installed in the buildings.

It has been shown that using collaborative comments and annotations can promote understanding of visual information [7]. Sense.us is a project for asynchronous collaboration in the context of information visualization. It is a web-based tool that supports view sharing, discussion and graphical annotation [15]. Envismo leverages a back-end system hosted by the Visualization Lab at UC Berkeley allowing interaction between mobile and desktop users.

Previous works by Yoo, Paelke and Chittaro explored the issues in visualization design on mobile devices due to the limited screen size and resolution [8][25]. Envismo brings focus to another limiting factor of mobile visualization, which is the computing resource and bandwidth of the device. Dix and Ellis [10] and Olken and Rotem [21] describe techniques to randomly sample data from a database. The approach we present in Envismo uses a simple random sampling technique to reduce the amount of data a mobile device needs to process in order to create the visualization.

# 4. ARCHITECTURE

An overview of our system architecture can be seen in Figure 2. The base system is a Nokia N810 Internet tablet to which we add additional sensors to the device using an attached Arduino microcontroller. A local datastore enables off-line data logging,

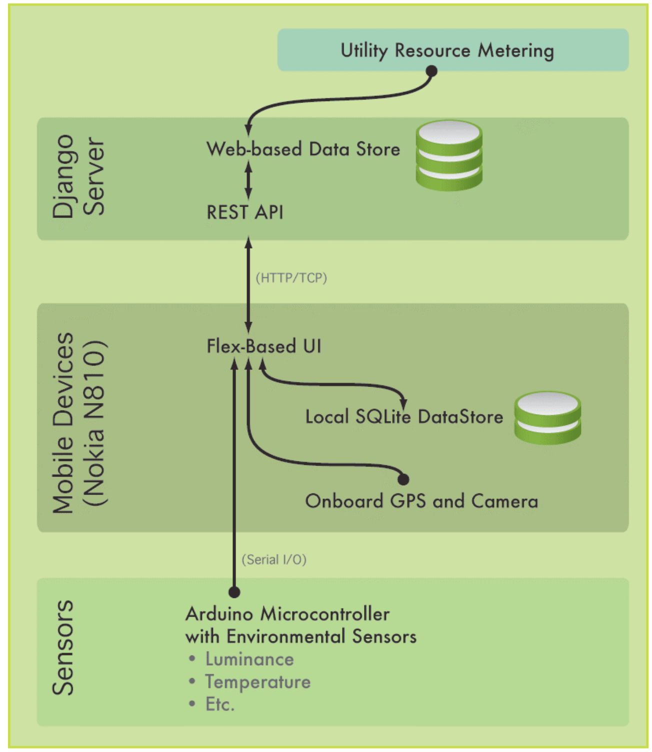


**Figure 2. Overview of the Envismo platform.**

while a webserver provides permanent storage and access to data from fixed resource meters.

## 4.1 Mobile Device

Location-awareness is an important building block of this kind of mobile system. Envismo supports location-aware data-collection and location-aware visualization by utilizing the N810's on board Global Positioning System (GPS) unit. We implemented a daemon process to communicate between the device hardware and the user interface via TCP. The GPS daemon listens for the change of GPS coordinates asynchronously. Each time the GPS device signal changes a call back in the daemon is evoked to handle the update. The callback function examines the new GPS coordinates from the device and compares it with the last received coordinates. The daemon only sends the new coordinates to the communication channel if the new location is more than a threshold distance away from the last transmitted location. This distance threshold is an optimization we made to limit the update frequency while maintaining functionality of the system. The GPS change signal may be triggered by several conditions such as change in the number of visible satellites or change in tracking speed but because only the altitude, longitude and altitude are relevant to Envismo only they are forwarded. Moreover, locations that are close to each other tend to have no detectable difference in sensor readings. Therefore, we filter out irrelevant updates to reduce the communication overhead. Upon receiving data from the communication channel the user interface updates the GPS coordinates display.

Envismo supports data logging even when there is limited or no Internet connectivity. This is accomplished through the use of a local SQLite database on the device. We decided to use a database rather than text files or some other as storage mechanism, because we wanted to be able to query data on the device in a manner similar to our web-based datastore (discussed later). We implemented a second daemon to handle requests from the user interface and execute them on the database. The database daemon asynchronously listens for input from the user interface and, upon receiving a request, parses it to determine the request type and contents of the request. The database daemon then uses the proper API call to execute the request. If SQLite returns a result set, the daemon converts the returned results into a JSON formatted string and hands it to the user interface through an asynchronous write to the communication channel. The JSON string is parsed and converted into objects by the User Interface.

# 5. SENSOR PLATFORM

In order to integrate additional sensors not built onto the N810, we augmented the tablet by connecting it via its USB port to an Arduino microcontroller. This provides a platform to which we can attach up to a dozen additional analog or digital sensors. It also provides with a processor capable of handling some initial data processing, scaling and calibration. The chief challenge here is that the Arduino writes data over USB to a serial port while our main application (described in Section 5) runs in Flash and is unable to read values from a serial port directly. A middleware or a serial proxy interface is needed to convert serial port messages to TCP messages that the Flash application can handle. Several existing solutions exist for moving data from a serially-connected Arduino to Flash including SerProxy [4] and AS3glue [6]. However, we struggled to find one that would work satisfactorily on the N810. After trying several stock approaches, we ultimately wrote a Python script and used it to redirect serial port messages to TCP sockets. The current pipeline for collecting and transmitting sensor data to the N810 is as follows:

1. Connect the sensors to the Arduino, and execute an Arduino application (a "sketch") that polls for analog and digital inputs and converts the measured values into standard units.

2. Set a symbolic link between the serial port (/dev/ttyS0 in our case) and the USB port (/dev/ttyUSB0). Without this link, reads from and writes to the Arduino conflict, causing the serial link to fail.

3. Switch the N810 into USB host mode so that it can power the Arduino[3].

4. Insert kernel modules into the mobile's OS to make its USB port appear as the serial port to the Arduino.

5. Execute the Python script (described above) to redirect serial messages to TCP.

6. Open a TCP socket in the Flash application and read the redirected TCP messages.

### *5.1.1 Thermistor*

To measure changes in temperature we use a thermal resistor ("thermistor") placed between a power pin and an analog input on the microcontroller. Using a lookup table in Arduino Sketch, (as explained in [5]) we translate input voltage into decimal degrees. The resulting degree data and units are then transmitted to the mobile device.

### *5.1.2 Luminosity Sensor*

We also incorporated a luminosity sensor in order to measure changes in visible brightness that might not be as accurately reflected by readings from the N810's onboard camera. This sensor required a substantially more complex conversion to translate raw voltages into accurate luminosity readings. These conversions were handled using a procedure detailed in [2].

### *5.1.3 Additional Sensors*

We utilized a thermistor and luminosity sensor here as a proof of concept. However, we could potentially use many of other types of sensors to gather additional relevant environmental data. Because the microcontroller is capable of reading any kind of analog input, any commercially available sensor that produces a varying voltage as output could be used. Table 1 lists a number or other potential sensor types along with possible applications for each.

**Table 1. Other Candidate Environmental Sensors**

| Sensor Type | Possible Uses |
|---|---|
| Light/Luminosity | Determine ambient light levels. Monitor interior and exterior lighting. |
| Temperature | Assess heating/cooling and ambient temperature. |
| Heat/Heat flux | Assess the heat generation/loss at a specific point. |
| EMF | Monitor the presence and strength of electromagnetic radiation. May correlate with electricity use. |
| Humidity<br>Pressure | Assess heating and cooling systems. Explore impact of weather in absence of a forecast network. |
| Odor<br>Carbon-Monoxide | May help detecting possible gas leaks, malfunctioning furnaces, etc. |
| Camera<br>IR Motion/Proximity | Monitor building activity and occupancy. |
| Acoustic | Monitor occupancy and noise levels. |

## 5.2 Web Datastore

Data drawn from utilities and data collected by mobile users are ultimately aggregated in a web-accessible datastore - in this case a Django-based web service with a relational database backend (MySQL). Utility data recorded at external metering points are pooled here and processed before being made available to mobile clients. The datastore also provides a central repository for sensor data collected by all clients. It is also intended to serve as a hub for collaboration, annotation, and data analysis, and provides access via a REST API to support additional applications.

### *5.2.1 Replicating Utility Data*

In many cases energy data from utility meters (if it is available at all) is already accessible via the web. However we found that an additional level of indirection was necessary in order to make loading that data on a mobile device tractable. In our case, data from the UC Berkeley Green Building Project is readily available through an API [23] that allows a client to query for all data on a single resource in a building over some time period (e.g. all energy usage data for Cory Hall, 2007-2008). Such a query could potentially return thousands (even hundreds of thousands) of datapoints, enough that even a desktop application might struggle to download and plot a query efficiently. Using our current mobile platform, we observed that downloading plotting more than a few hundred points resulted in unreasonable interactive delays. Additionally, a client may often want to query for data from a number of sensors simultaneously (as when generating a map view) but accessing the existing API to do so would require a string of independent requests (one per building), which is also problematic from a bandwidth- and resource-constrained mobile device. The current system addresses these issues by replicating data available from utilities on the Envismo datastore and providing a more targeted set of access methods suitable for mobile devices. For our prototype system we obtained a complete dump of a recent version of the full UC Berkeley energy usage database and loaded it manually into our own database. We then explored supplementing that data by regularly polling the available API and apppending new results to our own database.

### *5.2.2 Mobile / Datastore Interaction*

Mobile devices interact with the server via HTTP, pulling new data each time a visualization is generated. Because the mobile device is so resource-limited, we have tailored the web database to provide small, manageable results sets when visualizing data. Because we are focused on providing glanceable overviews of environmental data rather than exhaustive displays of all data, we use random sampling and aggregation to return small, but representative datasets. For some visualizations, including the map plots seen in our mobile application, require only a single reading be returned per sensor. In these cases we simply group by the sensor or device and return the maximum, minimum, or average value for each over a given time period. This produces a result set that is bounded by the number of sensors, which in our case remains a relatively small number.

Other plots, such as our timelines showing electricity consumption for a single building over several years, correspond to hundreds of thousands of datapoints in our web datastore. We leverage Dix and Ellis's observations [10] that, because Gestalt perceptual processing depends predominantly on approximate (rather than exact) properties of the visualized data, a random sample is often capable of visually conveying a dataset with high fidelity. To minimize the total transmission and rendering cost of these queries, we index the datapoints in our relational database by a random id and, when a query is executed, return only the first n results given this random ordering. We based our choice of n on observations that mobile performance deteriorated when attempting to plot more than a few hundred points. Moreover, on a small display (4.13" diagonally with 800x480 resolution) more points became difficult to distinguish and added little to the overall interpretation of trend data. In light of this we limit the number of results for these types of queries at roughly n=~200. Random sampling is not without its downsides. Maxima, minima, and outliers that might be accounted for by a more complex algorithm can either be missed entirely or can incorrectly appear to dominate, however, the overall impact can be less destructive than a systematic sampling technique and the overall roughness of the result tends not to suggest specific patterns in the results. Randomly selecting points also implies that two identical queries can produce different sets of results and, thus, our system may produce two different visualizations. Additionally, we acknowledge that while our methodology for selecting the cutoff is somewhat unscientific, anecdotally it seems to deliver a good compromise of resolution and load time. Moreover, because the ratio of selected to unselected points increases as the total number of query results becomes smaller, queries over smaller regions and smaller slices of time will return increasingly more complete results. This means that users intent on obtaining extact readings can always drill down to obtain them.

For both randomly sampled and aggregated queries, our approach pushes the bulk of the selection and computation away from the mobile device and on to the server, where additional resources and a more full-featured relational database are available.

We also tailor data upload to reflect the 'sometimes-on' nature of mobile connectivity. Data recorded on the mobile device is stored locally and passed in batch to the server when an Internet connection is available. This is important because the mobile devices need to be able to support data recording even in the absence of a wireless connection. In the current model, data is transmitted to the server via HTTP and assumes TCP/IP reliability guarantees. The current version of the system retains all locally recorded data permanently. However, in the future, the system could be augmented to remove data from the local datastore after a successfully acknowledged upload to the server. Because of the datalogging nature of the system, all writes to the datastore are simply append operations, and we need not be concerned with the possibility of conflicting writes.

### *5.2.3 Collaboration and Extensibility*

While our current build of the server is largely relegated to storing and serving for simple visualizations, we harbor a broader vision for the server in which it serves as a hub for collaboration, annotation, and data analysis. To this end we've attempted to provide access to all of the data stored on the server through a REST API [13] Currently this includes HTTP GET and POST methods for pulling and pushing raw data to and from the repository. In order to support collaboration, dialogue, and shared sensemaking around the data, we also imagine exposing the tools to add comments and annotations to data points and parameterized views as done in sense.us [15].

Metering or environmental data from additional outside sources can be added via calls to the API. While our current example system would actually poll an outside data source in order to obtain current data we allowing outside sources to push new data directly to our server via the API. We envision that a simple common interface for adding data to the system will make it easier to utilize in a number of settings, including a home or office, where access to real-time metering is becoming more prevalent. To add data to the system, an outside provider must make an API call containing the arguments provided in Table 2.

While all API calls to the database made from the N810 return aggregated or sampled readings, exposing access to all of the raw data is important for desktop and web-based applications. While the mobile component of the system is an important one, particularly for data collection, mobile devices are too resource- and I/O-limited to facilitate detailed analysis. This means that while collection, quick observations, and some commenting and annotation my happen on a mobile platform most of the real analysis will need to happen elsewhere on a more full-featured platform. In that light, we see the server and API as perhaps the most important component of the system, since it supports the application and synthesis of the collected data in a format where it may better facilitate insight and ultimately impact energy usage.

**Table 2. Arguments for calls to open data loading API http://{API URL}/energdb/readings)**

| Field | Description |
|---|---|
| date | Date and time of the reading |
| latitude | Latitude of the reading (degrees) |
| longitude | Longitude of the reading (degrees) |
| altitude | Altitude of the reading (meters) |
| sensorType | The type of sensor being used (e.g. "Electricity", "Temperature", etc.) |
| deviceId | A unique identifier (name) for the device. |
| sensorId | A unique identifier for a sensor used to make the reading. |
| sessionId | A unique identifier for the current recording session - establishes a relationship between points recorded together. |
| value | The numerical value of the reading (floating point) |
| units | The units in which the reading is returned. |

# 6. INTERFACE / INTERACTION

From the outset, we attempted to designed our application's user interface to facilitate a few specific interactions:

- Provide map-based displays of environmental data for a user's current location.
- Allow users to quickly fine tune the parameters of the current display to observe data from other time periods and locations.
- Make the process of recording additional data easy.

We sought to take advantage of the mobile platform, using geographic information from the onboard GPS where possible and providing glanceable displays that could be used as quick references for the user in situ. In general, this led us to create very simple plots and charts, plotting only one variable at a time and maintaining a relatively sparse layout. These visualizations are intended to serve as references for quickly appraising data and spotting trends. They are explicitly not designed to facilitate full-scale data analysis.

## 6.1 Map Visualization

The basic map visualization seen in Figure 3 serves as a user's entry point to the application. Unless otherwise configured, this view is designed to immediately center on the user's current location as indicated by the onboard GPS and display a map overlaid with average energy usage statistics for the nearby area. A row of controls along the upper edge of the application doubles as a title for the current view and a filtering control that allows users to select different time ranges, locations, and resources. In the default mode, where location is tied to the user's current GPS location, a slider below this title bar allows users to vary the radius around their current location for which data should be displayed. This allows users to fine-tune the visualization in order to have it display information from a specific area (alcove, room, building, etc.) in the absence of any system knowledge of the topography or building layout.

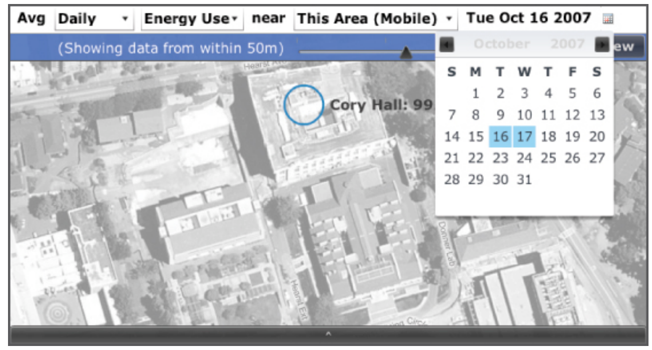


**Figure 3. Envismo's default map visualization highlighting electricity use for buildings in the local area over a 24-hour span.**

## 6.2 Line Chart Visualization

For any given resource or observation type and location, the system can also display a line chart visualization showing changes in value over time. An example of this view is seen in Figure 4. From this view, the title bar allows users to quickly toggle between 'daily', 'weekly', 'yearly' and 'all time' versions of the graph that can be useful for detecting trends and oscillations on varying temporal scales.

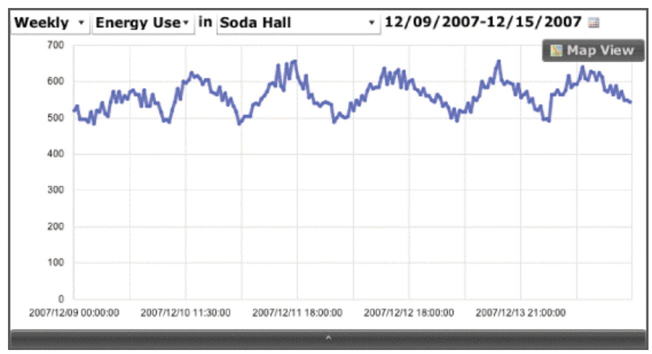


**Figure 4. A line chart visualization showing electricity use over the course of a week.**

## 6.3 Data Recording

Data recording controls are accessed via a foldout panel at the bottom of the display (Figure 5). This panel displays readings from the currently attached sensors as well as current GPS location data.

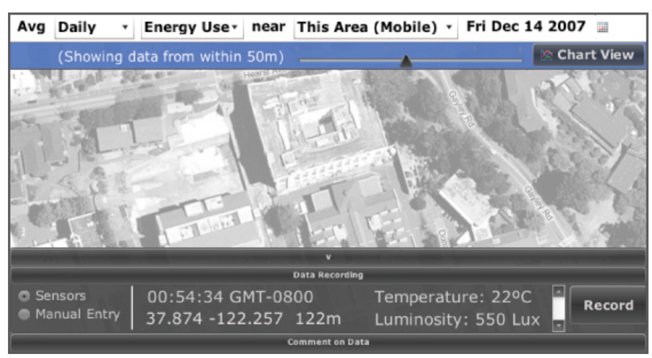


**Figure 5. User interface with the data recording panel expanded to show current sensor readings.**

Data recording from the sensors can be toggled on and off using a button at the right side of the panel. An alternate manual entry mode is also supported in order to allow users to input data for which they do not have a mobile sensor. Readings from a home energy meter or Kill-A-Watt, for example, might be particularly relevant to a user attempting to track usage at an appliance level.

## 6.4 Annotation

Similarly, a second fold out panel (visible in Figure 6) allows users to add text annotations attached to a particular time and location being viewed. Annotations entered on a view would be attached to the state of the current view, time stamped and pushed to the remote database where they could be accessed via the API and used in later analysis. While we implemented the user interface for adding comments in our current build, there is currently no way to view previously entered annotations on the mobile device and entered comments are not propagated to the remote database.

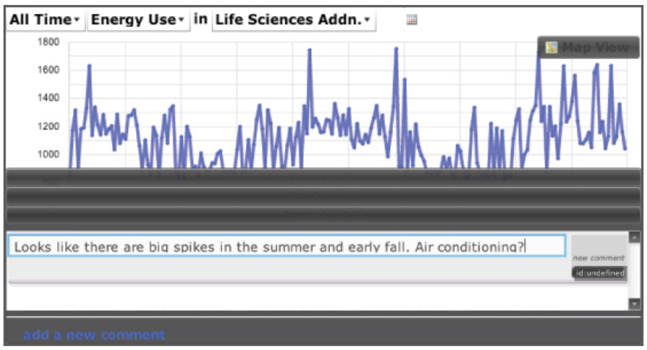


**Figure 6. The commenting panel, seen at the bottom of the interface, allows users to annotate the current view.**

## 6.5 Implementation

The user interface is implemented in Flash using the Adobe Flex GUI toolkit and runs within the N810's MicroB web browser. Visualizations were constructed using the Flare [14] visualization toolkit and utilize the Modest Maps [20] library for rendering map views. We selected this platform based largely on the relative ease with which it allowed us to iterate and interactively prototype the interface and visualizations. We were aware that this strategy introduced a considerable amount of overhead (Flex libraries on top of Flash on top of MicroB on top of a Linux-based OS is an auspiciously large application stack for a mobile device). However, we were encouraged by the CommScape [17] team's charitable description of Flex on the N810 as "slow but usable." In our experience this has generally been the case, with the application exhibiting passable but slower-than-desired performance. The most expensive operations typically involve loading map tiles, which can take from 10 to 15 seconds for a given view. These waits forced us to remove the ability to drag and zoom the map arbitrarily from the system after poor initial tests. However, we were able to achieve reasonable interactive speeds for all other major features. Certainly a final version of the application would benefit from a move to another platform, with Python and Flash Lite as viable options.

# 7. EVALUATION

While a user evaluation was not within the scope of our current project, we have discussed a number of potential studies that we might run given additional time. While the specific methodology of our user testing would depend more specifically on the results of our pilots, we propose a basic initial evaluation of the system along with more elaborate field trials.

## 7.1 Initial User Evaluation

Our evaluation would seek to vet the user interface and gauge participants' comfort with the system during a short (~ 1hr) user test in a controlled environment. Each participant would be asked a series of initial questions to determine their familiarity with existing energy monitoring technologies and mobile devices. Though the possibility of a user having used a system like ours on a mobile device is low, exposure to existing web applications like the Berkeley Dashboard is certainly a possibility.

Next users would be presented with the actual Envismo package and asked to interact with it. Participants would be asked to accomplish some predefined tasks in order to build familiarity with the system and test general ease of use. In the second half of the usability study, the user will be asked to interact with the system freely. After using the system, participants would be debriefed on their experience and asked to provide additional feedback.

## 7.2 Field Trials

Following a controlled validation of the prototype, we would also like to attempt a more qualitative study of actual use. This might entail providing sensor-equipped devices to several participants and asking them to use the system in over the course of several weeks. Users would be asked to actively use In this case it would useful to include participants who align strongly with our initial use cases, including students, researchers, and building managers on the Berkeley campus. This kind of study would be more likely to uncover less superficial problems with the system and help build a larger pool of environmental data for analysis. Over the course of such a study we would log usage on the mobile device and web service. We might also attempt to collect qualitative measures of the system using a diary study and recurring interviews. In the longer term, it would be interesting to conduct a larger study with significantly more participants in order to assess the collaborative potential of the system as well as its impact on participants' energy use.

# 8. DISCUSSION AND FUTURE WORK

We believe that our current system provides an interesting first step towards a viable system for environmental data collection an real-time glanceable usage visualization. However, it is still far from a complete system. While we have successfully demonstrated the potential for attaching remote sensors to a mobile device cheaply, a final solution would need to be much more polished. Ideally a sensing package containing a range of sensors (like those discussed in Section 4) could be provided in a small, self-contained unit that could be attached directly a mobile device. It seems likely that such a package could be produced relatively cheaply using off-the-shelf electronics similar to the ones used in our prototype. Such a package would should require minimal configuration and should be a "plug-and-play" piece of hardware that consumers simply need to connect to their device and begin using. In the less immediate future, as these sensors become increasingly inexpensive, they may even be integrated into the mobile device itself.

While the user interface permits users to visualize data from a number of sources, it does a relatively poor job of visualizing data from multiple sources simultaneously. This was a conscious decision early on, since it seemed useful to keep charts on the mobile from becoming crowded, confusing, or unreadable. However, doing so prohibits users from easily making quick comparisons between related variables and discerning correlations or trends. A balance needs to be struck between a flexible system that overwhelms users or is difficult to control via a mobile interface and the flat, oversimplified visualizations seen in the current version. One solution might be to develop a number of simple visualization types (similar in scope to the visualization components used by Many-Eyes [19] each with tightly scoped interactivity, for doing specific types of comparisons.

In the current system we have implemented some of the basic building blocks for tracking data provenance including unique ids for authors, devices, sensors, and recording sessions. However, our development version makes little use of this information, other than to group sets of recorded data points into session tracks. In a future system, however, this data could be used to filter out data from problem users or irrelevant sources. It could also be used to make assumptions about the topography and segmentation of the space in which readings are taken, or could be used to track a single user's environment and estimate energy usage over the course of a day.

On a related note, the current system makes no attempt to implement a security or authentication model of any sort, instead placing data from all users in the same unrestricted repository. A more complete system would need to consider the privacy of individual users more carefully, in no small part because of the presence of location and time information in mobile readings. Meter readings and billing data are also a potential concern since they ostensibly reflect a user's finances. This might mean anonymizing user data or even abstracting individual environmental observations so as to obscure the location of users at any given time. It might also entail segmenting the community into trusted groups or providing different datastores for individuals and groups who are uncomfortable with opening their data to others. Even an open system would need to take measures to prevent malicious users from inserting incorrect, misleading, or junk data.

In general, we believe a more social platform for analysis, sharing, and even competition is necessary in order to for this data to have an impact. This implies not just providing secure groups for controlling data access, but creating a richer, more social platform both on the mobile device and on the web in which this information could be explored and acted upon. This might entail tighter integration with a web-based data analysis platform built on top of our existing datastore. This might also include adding a social component to the system or integrating with existing social networks in order to further engage the audience.

Finally, we emphasize that we imagine this application applying to a much wider range of potential applications beyond the campus-wide scenario presented in our examples. Such a system might be even more useful and personally relevant to users in a small office or home setting where an individual's potential to affect a measurable change is substantially greater.

## 9. ACKNOWLEDGMENTS

We would like to thank Prof. Brewer for the insight and guidance he provided us as we were selecting our project. Moreover, the course as a whole was interesting, clear, and generally well taught. Thank you!

## 10. REFERENCES

[1] Arduino. http://arduino.cc/

[2] Arduino and the Taos TSL230R Light Sensor: Getting Started. http://roamingdrone.wordpress.com/2008/11/13/arduino-and-the-taos-tsl230r-light-sensor-getting-started

[3] Arduino Interaction: Location-aware mobility. http://blog.locationist.net/2008/03/16/arduino-interaction

[4] Arduino Playground: Serial Net. http://www.arduino.cc/playground/Interfacing/SerialNet

[5] Arduino playground – Thermistor. http://www.arduino.cc/playground/ComponntLib/Thermistor

[6] As3Glue. http://code.google.com/p/as3glue

[7] Brodlie, K. W., Duce, D. A., Gallop, J. R., Walton, J. P. R. B., and Wood, J. D. 2004. Distributed and collaborative visualization. Computer Graphics Forum 23, 2, 223--251.

[8] Chittaro, L. 2006. Visualizing Information on Mobile Devices. Computer 39, 3 (Mar. 2006), 40-45. DOI= http://dx.doi.org/10.1109/MC.2006.109

[9] Danis, C. M., Viegas, F. B., Wattenberg, M., and Kriss, J. 2008. Your place or mine?: visualization as a community component. In Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 275-284. DOI= http://doi.acm.org/10.1145/1357054.1357102

[10] Dix, A and Ellis, G, 2002, "By-chance - enhancing interaction with large data sets through statistical sampling" Proceedings of the Working Conference on Advanced Visual Interfaces 167-176.

[11] Ellis, S. E. and Groth, D. P. 2004. A collaborative annotation system for data visualization. In Proceedings of the Working Conference on Advanced Visual interfaces (Gallipoli, Italy, May 25 - 28, 2004). AVI '04. ACM, New York, NY, 411-414. DOI= http://doi.acm.org/10.1145/989863.989938

[12] Energy Use @ MIT. http://energymap.mit.edu/map

[13] Fielding, R. T. and Taylor, R. N. 2000. Principled design of the modern Web architecture. In Proceedings of the 22nd international Conference on Software Engineering (Limerick, Ireland, June 04 - 11, 2000). ICSE '00. ACM, New York, NY, 407-416. DOI= http://doi.acm.org/10.1145/337180.337228

[14] Flare. http://flare.prefuse.org/

[15] Heer, J., Viégas, F. B., and Wattenberg, M. 2007. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In Proceedings of the SIGCHI

Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1029-1038. DOI= http://doi.acm.org/10.1145/1240624.1240781

[16] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S. 2006. CarTel: a distributed mobile sensor computing system. In Proceedings of the 4th international Conference on Embedded Networked Sensor Systems (Boulder, Colorado, USA, October 31 - November 03, 2006). SenSys '06. ACM, New York, NY, 125-138. DOI= http://doi.acm.org/10.1145/1182807.1182821

[17] K. Chen, K. Heimerl, C. Robson, B. Trushkowsky. CommScape:Information Visualization, Collaboration and Delivery for Rural Health Care. Unpublished.

[18] Live Building. http://livebuilding.queensu.ca

[19] Manyeyes. http://manyeyes.alphaworks.ibm.com

[20] Modest Maps. http://modestmaps.com/

[21] Olken, F. and Rotem, D. 1986. Simple Random Sampling from Relational Databases. In Proceedings of the 12th international Conference on Very Large Data Bases (August 25 - 28, 1986). W. W. Chu, G. Gardarin, S. Ohsuga, and Y. Kambayashi, Eds. Very Large Data Bases. Morgan Kaufmann Publishers, San Francisco, CA, 160-169.

[22] P.M. Aoki, R.J. Honicky, A. Mainwaring, C. Myers, E. Paulos, S. Subramanian, and A. Woodruff. Common Sense: Mobile Environmental Sensing Platforms to Support Community Action and Citizen Science (demonstration). Adjunct Proceedings, Ubicomp 2008, Sep. 2008, 59-60.

[23] UC Berkeley Building Energy Dashboard. http://www.demandless.org/building/

[24] Wood, J., Wright, H., and Brodlie, K. 1997. Collaborative visualization. In Proceedings of the 8th Conference on Visualization '97 (Phoenix, Arizona, United States, October 18 - 24, 1997). R. Yagel and H. Hagen, Eds. IEEE Visualization. IEEE Computer Society Press, Los Alamitos, CA, 253-ff.

[25] Yoo, H. Y. and Cheon, S. H. 2006. Visualization by information type on mobile device. In Proceedings of the 2006 Asia-Pacific Symposium on information Visualisation - Volume 60 (Tokyo, Japan). K. Misue, K. Sugiyama, and J. Tanaka, Eds. ACM International Conference Proceeding Series, vol. 164. Australian Computer Society, Darlinghurst, Australia, 143-146.