The Software Design of an Intelligent Water Pump

Daniel Scott Weaver(✉) and Brian Nejmeh

Department of Computer and Information Science,
Messiah College, Mechanicsburg, PA 17055, USA
sweaver@messiah.edu

Abstract. In an effort to increase handpump reliability, the Messiah College Collaboratory and the Department of Computer and Information Science are developing the Intelligent Water Project (IWP) system. IWP measures and reports the functionality of handpumps and volume of water extracted on two-hour intervals daily. Additionally, IWP will measure groundwater levels which can be used to evaluate well yields. Data from handpumps is automatically collected and transmitted to a remote database. Once in the database, the data is analyzed and distributed to stakeholders via web and mobile applications and customizable alerts. Besides monitoring water extraction, handpump performance, and borehole health, the IWP system processes data to alert stakeholders of failure or degrading conditions (imminent failure). Coupled with appropriate field management processes, this information can lead to improved handpump availability and lowered cost of ownership. The key goal is to dramatically increase the reliability of handpumps. A secondary goal is the collection of handpump data from all IWP enabled pump sources providing a rich resource of data to enabling WASH practitioners, managers, hydrologist and donors to make more informed decisions. The purpose of this paper is to highlight the key IWP software design considerations and to discuss the key software design decisions made and the rationale for the same.

1 Introduction

Wells and handpumps in Africa fail at alarming rates within the first two years of installation [1]. Much of this failure can be attributed to a lack of transparency into the performance of handpumps. Existing manual methods of handpump monitoring require manual field inspection by personnel which is costly, untimely and superficial. Furthermore, traveling long distances to reach handpumps results in infrequent inspections.

The advent of low cost, reliable sensor technology coupled with the ubiquitous GSM network has the potential to bring unprecedented levels of transparency to handpump performance in rural Africa. Our project has refined a fully automated wireless, sensor-based mobile and web application suite to provide significant remote transparency of the overall handpump performance.

C. Grueau et al. (Eds.): GISTAM 2016, CCIS 741, pp. 164–180, 2017.
DOI: 10.1007/978-3-319-62618-5_10

Initial concept development of the Intelligent Water Project (IWP) sensor technology and the software suite began in 2012 with internal funding from the Messiah College Collaboratory and the Department of Computer and Information Science (CIS) with subsequent funding from World Vision. This project is being done by a coordinated group of faculty members and students across various engineering and computer science disciplines at Messiah College, a Christian college based in the United States. The software for this project has been developed using the Agile Scrum method [2] in two service-learning computer science classes (database applications, senior capstone course in CIS). Given the Christian-faith tradition of Messiah College, we often use Biblical references to help motivate our work. Work on the IWP project has been inspired by the following passage: "*The poor and needy seek water, but there is none, their tongues fail for thirst. I, the Lord, will hear them; I, the God of Israel, will not forsake them. I will open rivers in desolate heights, and fountains in the midst of the valleys; I will make the wilderness a pool of water, and the dry land springs of water.*" Isaiah 41:17–18 (NKJ).

A system-level overview and results to date of the IWP, including the overall hardware and communications design, were chronicled in a prior paper [3]. This prior paper also contained a broad review of related work. The focus of this paper is on the software-specific design considerations and decisions made in the IWP.

IWP is differentiated from other handpump monitoring systems because it:

1. provides support for automated, sensor-based handpump data collection over the ubiquitous GSM network,
2. provides full transparency and access to all of the underlying sensor data via the website,
3. supports configurable, periodic status alerts on user defined events of interest,
4. leverages the work of the Messiah College India MKII and Afridev Sustainability Studies that gives unique insight and focus to the sensor design [4],
5. provides full integration with Google Maps® and ESRI (GIS cloud environment) systems,
6. is a cloud-based application suite which runs on desktops and mobile devices,
7. is being developed by an interdisciplinary team of hydrologists, mechanical engineers, electrical engineers and computer scientists.

2 Key Requirements

2.1 Problem Statement

Approximately 184 million people living in Africa depend on handpumps for their daily water supply [5] with an estimated 50,000 new handpumps shipped to Africa each year [6]. Despite efforts to improve rural water service delivery, handpumps serving rural communities often fall into disrepair. According to data compiled by Rural Water Supply Network (RWSN) [1] from 20 African nations covering 345,071 wells in 2009, 36% of handpumps are non-operational. This results in a loss of capital investment in infrastructure and a negative impact on rural communities. When a community handpump breaks down, families are forced to find alternative water sources. Alternative sources may include carrying water a greater distance from a handpump in a neighboring community, or less protected sources such as hand dug scoops or surface water. The latter sources carry increased risk of water born disease. The increased time and energy spent collecting water and the potential for illness detract from more economically empowering activities.

Logistical challenges and costs hamper effective and efficient handpump monitoring and evaluation efforts in rural areas. To determine the condition of a handpump, water authority representatives must travel to each handpump location and perform a manual inspection. This process can result in lengthy down-times and high labor and transportation costs incurred by the community and/or sponsoring NGO or government organization. As a consequence, handpumps may go weeks without necessary repairs and Water and Sanitation Hygiene (WASH) managers are forced to make critical program decisions on incomplete data.

Given the critical importance of clean water, it follows that an accurate, reliable and low-cost tool to assess handpump performance efficiency and effectiveness would be valuable to many stakeholders. Improved handpump transparency can lead to better visibility and early warning of handpump problems. This will enable timely handpump remediation, thereby leading to improvements in overall pump efficiency and effectiveness in service to rural African communities.

2.2 Solution Overview

The primary goal of IWP is to develop a system to automatically capture and organize data about handpump functionality and performance from both sensor and human sources. This allows the IWP to alert stakeholders via web and mobile app, email, and text messaging of pump failure or degrading conditions. Coupled with appropriate field response processes, the information the system provides can lead to improved handpump availability with a lower cost of ownership. A secondary goal is the collection of handpump data from all IWP enabled pump sources providing a rich resource of data to enabling WASH practitioners, managers, hydrologists and donors to make more informed decisions.

The IWP team decided on the following design goals and desired outcomes to drive our process:

Design Goals

1. Design a solar-powered, GSM-enabled, pump monitor with an array of sensors to communicate with a cloud-based database application,
2. Design a web-based application suite to produce actionable information about handpumps,
3. Design a mobile app that exploits location-aware and other mobile capabilities for local field technology workers.

Desired Outcomes

1. Improved visibility of handpump performance and ease of maintenance and reporting,
2. Improved understanding of water extraction for each handpump (how much and when),
3. Improved understanding of well water level fluctuations,
4. Single, unified source for storage, access, and analyses of handpump related data.

3 Architecture

The IWP remotely monitors handpumps, including the Afridev and India MKII, through the use of an embedded monitor installed in the handpump. The monitor, connected to and collecting data from concurrently installed sensors, is equipped with a GSM modem to communicate with the cloud-based database application via text messaging through an SMS receiver service. The cloud-based database application parses the transmitted data, populates the database and determines the performance and status of the handpump (See Fig. 1).

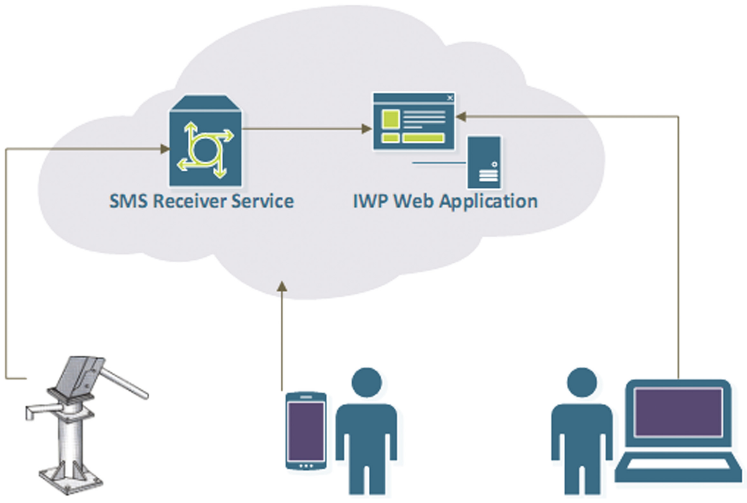

Fig. 1. The conceptual overview of IWP.

Each day the IWP embedded monitor measures and records the volume of water extracted by the handpump in two hour intervals, the amount of effort required to prime the pump, and the largest recorded leakage rate. This automatically collected data is transmitted daily to the remote database where the information is analyzed and made available to stakeholders via a web or mobile app. In addition to monitoring water extraction, handpump performance, and borehole health, the IWP system processes this data to predict certain degrading conditions before failure occurs and notifies stakeholders via customizable alerts. For instance, an increasing amount of effort required to prime the handpump may indicate degradation of handpump parts while decreasing yield may indicate silting at the screened interval.

In the event of an immediate handpump failure or degrading condition, the system automatically generates email and/or text message notifications to community members and area handpump technicians, mobilizing them to inspect the handpump and make repairs. Certain known failure modes are detected and reported by IWP, enabling repair teams to carry the needed parts, supplies, and tools to the site. Community members and area handpump technicians will also have the ability to report data such as cost of repairs or other visible handpump problems using the mobile app. Once a handpump is repaired, sensor data will verify handpump performance and close the failure reporting loop.

IWP is segmented into five architectural areas each with design goals and outcomes, providing team focus.

Monitor. Development of robust hardware powered by the sun that collects sensor data in 2-hour increments, formats a GSM text message with valid JSON formatting from the sensor data, and transmits the message to the database application. The Monitor, mounted inside a handpump, must be uniquely identifiable.

Data Transmission. Develop a mechanism to allow communication between uniquely identified pumps and the IWP system so that data can be collected to be viewed and analyzed in the web interface. The design must include the ability to communicate with the monitor, providing the ability to retransmit or dequeue messages.

Database. Develop a database model that is configurable and flexible to handle different requirements of different climates and landscapes from which the pumps send data.

Web Application. Inform the general public about IWP through blog entries and allow authorized users to access critical pump data through a mobile friendly web interface. Develop an intuitive graphic user interface through which users select pumps or pump groups via a geographic map and select data and graphs representing the sensor data from the pumps.

Mobile Application (Mobile App). Provide field technicians the ability to connect to the IWP system while in the field with recognizing that technicians will be in areas where 3G connectivity is unavailable.

4 Design Considerations

The main software subsystems of IWP include the monitor hardware, database, data transport, mobile app, and web interface. This section of the paper highlights the key design considerations for each of the IWP software subsystems.

4.1 Monitor Hardware

The IWP system hardware consists of a solar-powered, GSM cellular-enabled sensor node that mounts inside of India MKII and Afridev handpumps. A prior paper [3] details the hardware and mechanical design of IWP on the India MKII platform. The system monitors the motion of the handpump handle and the presence or absence of water in the mouth of the rising main to (a) ascertain the amount of upstroke required to prime the pump, (b) the amount of water extracted from the pump and (c) the rate of leakage in the rising mains. This information is summarized and sent daily to the IWP database.

The IWP monitor is embedded software that runs on a processor mounted inside of the handpump. It manages the collection, summary calculations and formatting of the sensor data. In essence, the monitor software collects, processes and formats the sensor data, thereby packaging the sensor data for transmission to the IWP application. There were several key software design decisions made with respect to the monitor:

1. global pump identification,
2. data granularity,
3. data format,
4. data structure.

The global name space for pump identity proved to be a significant design consideration. Given the hierarchical nature of pump identification among organizations using pumps (i.e., country, region, local community identifier), at first we thought we could leverage such information and create a hierarchical name space. However, such names were often quite lengthy in characters and could contain non-ASCII characters. A second option was to simply use the phone number of the SIM card used by the monitor to transmit the sensor data. However, a complication of using such a phone number is that it is entirely possible to relocate a SIM card to another pump for data transmission. Doing so would mean that the same SIM card could be associated with more than one pump over time. We also felt it was better for the global pump identity to be a property of the pump itself and nothing else. For this reason, we used a system generated global unique numerical identifier for global pump identity.

A second major design decision concerned the granularity of sensor data to be managed by IWP. For example, should we record the volume of water extracted at a pump every hour, every eight hours or every twenty-four hours? One could ask a similar question about the leakage of water within a pump. In the end, we decided to store the following data about a pump on a daily basis:

1. the daily average of water lost through leakage when the rising main is full of water,
2. the daily longest amount of upstroke required before water starts to be extracted (i.e., so-called "longest prime"),
3. estimated volume of water extracted (adjusted for leakage) per 2 h interval,
4. average extraction upstroke per pumping event.

A third significant software design consideration involved the format used to store and transmit the sensor data. We wanted a format that was an industry-standard and commonly used. The format also needed to be efficient (i.e., little data header and other overhead). In the end, we decided to use JSON [7], a widely used industry standard attribute-value format for our sensor data. In short, JSON met all of our data format criteria.

Finally, consideration had to be given to the data structure used to store and transmit the sensor data. Given the attribute-value pair nature of JSON, we decided to use a simple one character attribute identifier for each sensor data attribute followed by the value of the data for that attribute. Given that near-time sensor data analysis and reporting was deemed sufficient by field water specialists we consulted, we decided that we could transmit all of the data associated with a sensor over a 24 h period (i.e., one day) in a single JSON message. In addition to the date, one additional data item was included in the JSON message. In order to determine if each JSON message transmitted was successfully received and processed by the IWP application, we added a unique message sequence number to each JSON message. The next section of the paper will discuss the data transport design and how the unique message sequence number was used to guarantee delivery of all JSON messages to the IWP application.

In the future, we will consider adding additional sensors, and in turn, sensor data to the IWP. This will be easily accomplished by extending the JSON attribute-value format structure to incorporate the additional sensor data. Finally, consideration will be given to using a binary encoding of the JSON objects to provide an extra level of security and greater capacity to transmit data in a single JSON object.

4.2 Data Transport

The data transport software layer is responsible for the successful transmission of the JSON encoded sensor data over SMS. The data transport layer software design is depicted in Fig. 2. The Sensor data collected by the monitor is stored on both an SD Card installed in the monitor and resident memory. Every twenty-four hours the monitor packages the collected data as a JSON formatted string and creates an SMS message. The data transmission occurs at a slightly different absolute time (after midnight local time) on all monitors to reduce any possibility of a transmission bottleneck from occurring. The system transmits the SMS message over a GSM (voice grade) wireless network to an SMS Receiver Service. It is commonly known and field studies in Africa have shown [8] that voice grade

GSM network service is much more widely available than 3G network service. Given the desire to field IWP in remote areas of Africa, the decision was made to only assume a voice-grade GSM network in our design.

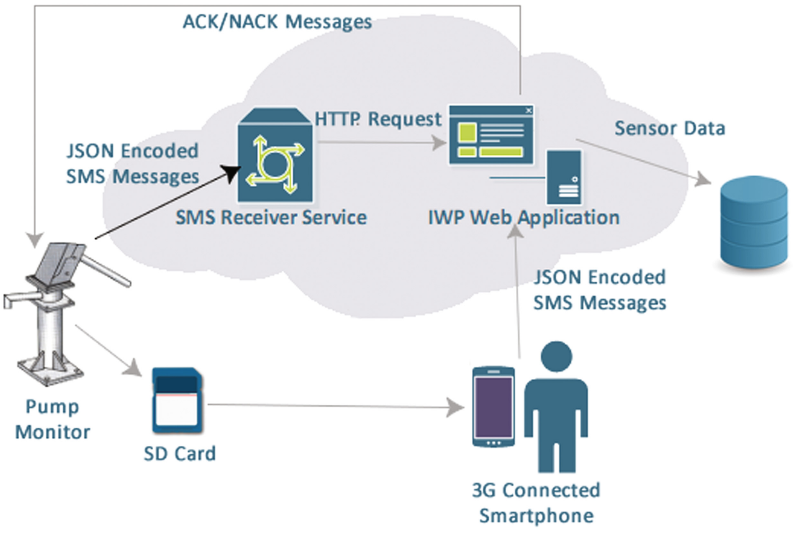

Fig. 2. Data transport.

The SMS Receiver Service forwards the message to the parser module in the cloud-resident database application. The parser module posts the raw message in the database, parses the message and (if error-free) stores the parsed sensor data in the database.

These SMS messages are equipped with unique sequence identifiers (USI). Once a message has been successfully processed and added to the database, an ACK message is sent to the monitor. Once received, the monitor will remove the message from its queue based on the USI.

If a message is missing data or the data does not conform to the defined data format, an ERR message is sent to the monitor. The monitor will then attempt to re-transmit the message.

When a duplicate message is encountered, a DUP message is sent to the monitor. In response, the monitor deletes the message from the queue.

There are several important future software design considerations for the data transport layer of IWP. To date, we have partnered with Upside Wireless® [9] to obtain a reliable SMS Receiver Service. In order to reduce costs and have greater flexibility, we are exploring the possibility of standing up and self-managing an open source SMS Receiver Service. Secondly, in some locations 3G+ wireless network service is reliably available. Such a wireless service would allow for the data transport layer to be IP-based and not require the use of SMS, but instead have the JSON data structure directly connect to the listening data parser on the server. Furthermore, such IP-based monitor software would allow for secure and direct monitor-based access to the sensor data on a more near-time basis (if that were desirable). Finally, consideration should be given to the digital replenishment of SMS credits for transmission of SMS messages by the monitor.

4.3 Database

The IWP data is housed in a secure, cloud-based database. The main components of the database are depicted in Fig. 3.

When an organization installs an IWP Monitor in a handpump within a community, the information necessary to track that handpump is stored in **Organization**, **Pump**, **Community**, and **Part**. The system administrator is then able to link that handpump to authorized **User**s who then have the ability to view information about that pump anywhere in the world. As soon as the Monitor is installed and operational, it begins its collection and transmission of sensor data to be stored in **Sensor Data** (as described in the previous section).

1. **Organization** provides the information related to the organization that is responsible for the installation of the pump.
2. **Pump** provides GPS coordinates, an organization-provided identifier, activation date, GSM phone number, and overall health data including the current status of the pump.

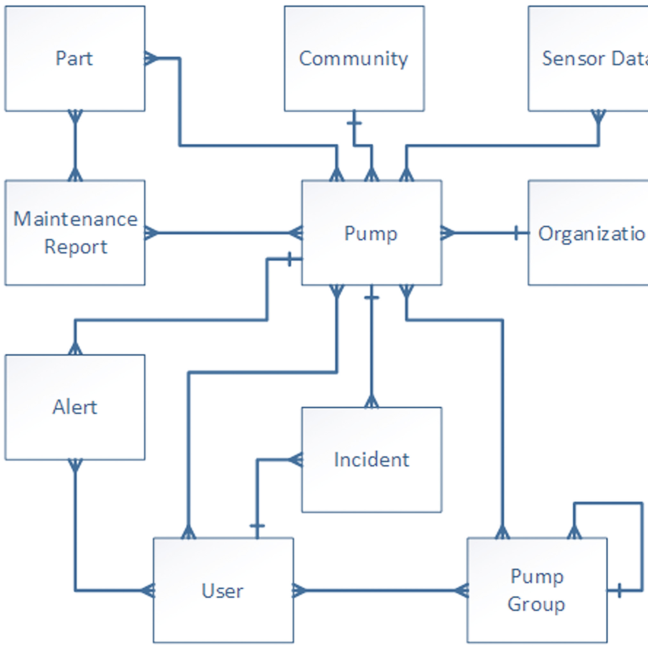

Fig. 3. Database conceptual model.

3. **Community** provides the information related to the community in which the pump resides including the community name and country.
4. **Part** provides information about the parts contained in the pump along with field maintenance records for replacement parts including commission and decommission dates, cost of part, and other maintenance-related information.
5. **User** provides the information for authorized users of the system including personal information, access level, and pumps they are authorized to monitor.
6. **Sensor Data** is stored both as un-parsed JSON strings as well as parsed JSON strings, providing access to all sensor collected data.

The insertion of sensor data into the database triggers a process that calculates health indicators, such as leakage rate and, based on configurable parameters, determines the current status of the handpump. If the status of the pump changes, the system creates an **Incident** report (SIR), which in turn generates an **Alert**. The user, having the capability to indicate how he/she would like to be notified, will receive the alert on login to the web or mobile app, email, or via text message.

Any authorized user may also create an **Incident** report based on their observation of the handpump. These human-generated incident reports (HIRs) are included in the calculation of the pump status. **Alert**s are generated based on the worse of the two types of incident reports (HIRs or SIRs).

1. **Incident** is divided into two categories: System Incidents and Human-recorded Incidents. System Incidents contain information related to the calculated health indicators of the pump. There are methods to customize what thresholds are used to change the status of pumps. Human-recorded incidents are those field technicians may enter via the mobile app. These are combined with System incidents to determine the overall health of the pump.
2. **Alert** is used to provide messages of health changes in a pump. The system is designed to allow authorized users to indicate the frequency of the sending of alert messages as well as the desired device through which the message is received.

IWP provides a mechanism for grouping handpumps together using **Pump Group**s. Pump groups allow for aggregate analysis, reporting, and searching across pumps in that group. Pumps can belong to more than one group. A pump group has a unique name and a brief description and allows pumps to be grouped by location, field technician, type of pump, or any user-defined grouping.

Pump groups were designed to provide an organizational tool for users. In the future, the implementation of global pump groups will allow users to easily view their assigned pumps within those groups. For example, pumps within the borders of a country would be assigned to that country pump group and users would be able to see their assigned pumps within that country without creating their own grouping.

An additional future consideration will be to automatically generate pump groups based on attribute-pair configuration parameters. For example, pump groups may be generated with all handpumps within an x-kilometer radius of a given GPS location, or those assigned to a given field technician, or within a certain geographic boundary such as a country.

When handpump technicians perform maintenance on a handpump, they complete a **Maintenance Report**, identifying their incurred travel, part, and labor costs. As mentioned above, maintenance reports also include **Pump** and **Part** data, providing a mechanism for calculating the cost of ownership for any given pump.

Security Model. Security within the database is modeled after a Roles and Permissions Matrix with the addition of pumps the user is authorized to access. When a user is defined to the system, a role is given to that user as well as authorization to pumps within the system. The role provides permissions for access to and manipulation of the pumps they are assigned. However, the system is also designed to allow the system administrator to modify specific permissions over specific pumps. For example, a field technician is authorized to add Human Incident Reports as well as Maintenance Reports, but is only allowed to view basic pump information. Suppose a field technician is a part of a community water committee. That technician may also be given access to the cost of ownership information for the pump within his community.

Current roles within the system include:

1. Administrator
2. Basic User
3. Donor
4. Field Technician

Data Triggers. A series of database triggers are executed with the initial insert of the un-parsed sensor data. As part of the parsing routine, the JSON string is validated and if it is determined that the JSON string is invalid, along with sending an ERR message to the monitor, the "bad data flag" is set in the Unparsed Sensor Data table. The insert of the un-parsed sensor data triggers a process to check the bad data flag and if it is true, insert data into the Historic Status table.

If the un-parsed data is successfully parsed, another trigger performs calculations on the sensor data and inserts the appropriate data into the Sensor Calculations table. That insert triggers yet another routine that derives status values based on the stored calculations and open HIRs and writes the data to the Historic Status table.

Periodically a script is run to check each pump for data in the Unparsed Sensor Data table. If no data is found for a configurable amount of days, the pump status is set to "grey" and data is entered into the No Sensor Log table. This triggers another routine which updates the status of the pump and inserts the data into the Historic Status table.

On any insert into the Historic Status table, another trigger is executed which determines if an alert needs to be sent. If so, the alert data is stored in the Pump Alert table and yet another trigger updates alerts for the appropriate users.

Triggers provide an automated method of identifying the status of pumps and sending the alerts to the appropriate users.

4.4 Web Application

The data from the monitor is collected and stored automatically in the database which can be accessed by authorized users via a secure web application or mobile app. The status of individual handpumps can be viewed on a map interface powered by Google Maps®. Each handpump on the map is represented by the pump status indicator (green, yellow, orange, red, or grey) depending on the level of functionality of the handpump.

The reporting module allows users to select a time period, single or multiple handpumps, or pump groups for further investigation, and provides either detailed or aggregated information. Selection can be accomplished via the map interface by selecting individual handpumps or pump groups. The IWP web application can export these queries as printable PDF reports or MS Excel Spreadsheets for further investigation or reporting purposes.

Notifications are handled automatically by the IWP software in the event of a change in a handpump status indicator. These are sent to appropriate stakeholders via email or text message depending on their preferences. The notifications are sent in the case of degradation, such as a status change from green to yellow or red, and in the case of an improvement, such as a status change from red to yellow or green. This allows stakeholders to know not only when a pump is broken, but also when and to what extent it has been repaired.

Technology Stack. The technology stack was decided upon with an effort to leverage open-source technology for the sake of Non-Profit Organizations who may not have large budgets. Therefore, the following was decided upon:

1. Server Software: Ubuntu version 12.0
2. Web Server Software: Apache version 2.0
3. PHP version 5.4.6
4. MySQL version 14.14
5. Smarty version 3.1
6. Bootstrap version 3.0.0
7. jQuery version 1.10.2

Application Structure. The Web Application was designed to abstract the presentation layer from the business logic layer and the business logic layer from the data layer. The presentation layer utilizes Smarty, a PHP Template Engine [10]. The data layer is encapsulated in PHP Classes. The business logic layer is written in php files within the public_html folder of the application. This decision was driven by the desire to insulate the application from database changes so that developers are unaffected by changes.

Reporting and Analytics. As mentioned in a previous section, the insertion of data into the database triggers a process that determines the current status of the handpump based on the current sensor data and human incident reports.

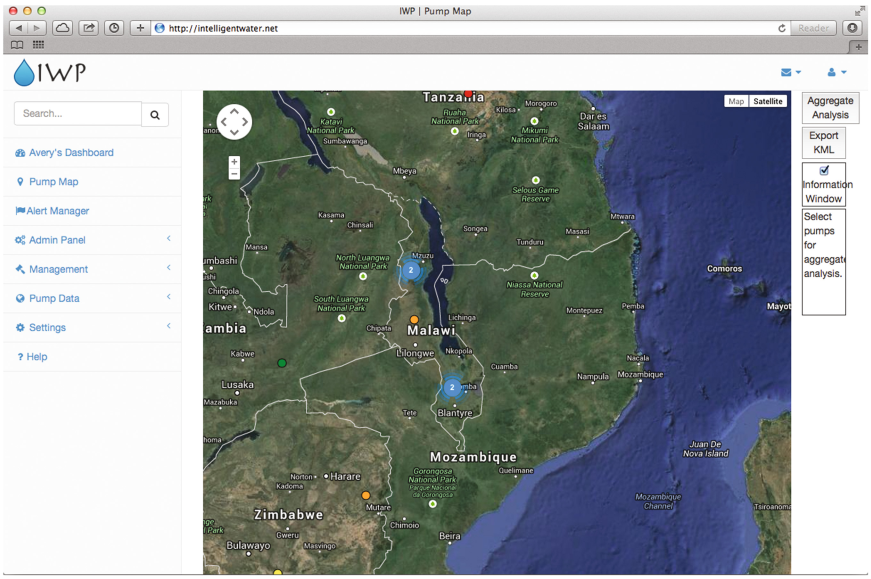

Fig. 4. Pumps on a Google Map®. (Color figure online)

A handpump will have the status of Green (running fine), Yellow (concerns exist about the handpump), Orange (significant problems exist with the handpump), Red (pump failure), or Grey (handpump has not been heard from). This status will be displayed on an authorized user's dashboard and the handpump location depicted on Google Maps® (See Fig. 4). If the status of the handpump changes, the alert system notifies the appropriate authorized users of the change. Handpump status is a configurable function defined for a given set of handpumps based on the values of daily volume extraction, leakage rate and maximum well level. An administrator has the authority to configure the Yellow and Orange statuses knowing that a pump more healthy than Yellow is Green and a pump less healthy than Orange is Red. Grey has been chosen for pumps from whom no sensor data has been received for a configurable number of days.

Google Maps® Integration. In order to make the web application user friendly, pumps were displayed as an overlay of a Google Map® utilizing the Google Maps® JavaScript API [11]. When a user logs in and selects the Pump Map, the general location of their pumps shows up on a Google Map® within the web application. The default zoom is applied and pumps are shown as circles in aggregated groups. As the user zooms in, the pump aggregation becomes individual pumps. The pump circles are colored based on the pump's status. Pumps may be selected from the map to view the Pump's profile.

On the Pump's profile page, graphs of the pump's leakage rate, longest prime, and volume extracted may be viewed along with the pump's basic data (Pump Name, Phone Number, Activation date, and GPS coordinates). The date range may be changed by the user to see specific days or ranges of days. Along with the graphs is a table of the detailed sensor data.

Graphs of multiple pumps at a time may be viewed by selecting multiple pumps on the map and choosing the aggregate view which will display the pump data together.

Along with the Google Maps® interface, we also support the ESRI system [12] by exporting a KML file using the Google Maps® API which in turn is imported into the ESRI system.

Some future mapping considerations are to allow other research organizations to overlay data with data found in IWP to get a better picture of what is happening in any given region.

4.5 Mobile App

A mobile app is an important element of the overall IWP system. It serves as a lightweight tool for workers to use while in the field, thereby enabling them to take advantage of mobile services such as location awareness and offline modes.

User Authentication: Mobile app users login to the mobile app using the same user name and password credentials of the web application. The mobile app authenticates users using the same web backend system that authenticates web users. Furthermore, the mobile app limits user access to handpump data and functionality identical to the security and access model imposed by the web application.

Mobile App Interface: Figure 5 displays the main screen of the mobile app. The core IWP mobile app functions are handpump initialization, filing maintenance reports, filing incident reports and viewing handpump alerts.

Handpump Initialization: This function allows a user to initialize a handpump into the IWP system. The function records the field technician assigned to the handpump, the GPS coordinates of the handpump (either automatically recorded (default) or entered by the user), the phone number on the SIM card installed in the handpump monitor, the date/time of the initialization and other descriptive information about the handpump.

Maintenance Reports: This function allows a user to create and submit handpump maintenance reports (as previously described), including the identity of the handpump, the user filing the maintenance report, date and time of the maintenance report, a brief description of the maintenance performed on the handpump and the total cost of the maintenance report broken down by travel, part, and labor costs.

Incident Reports: This function allows a user to create and submit handpump incident reports (as previously described), including the identity of the handpump, the user filing the incident report, date and time of the incident report, a brief description of the incident being reported on the handpump and the nature of the incident.

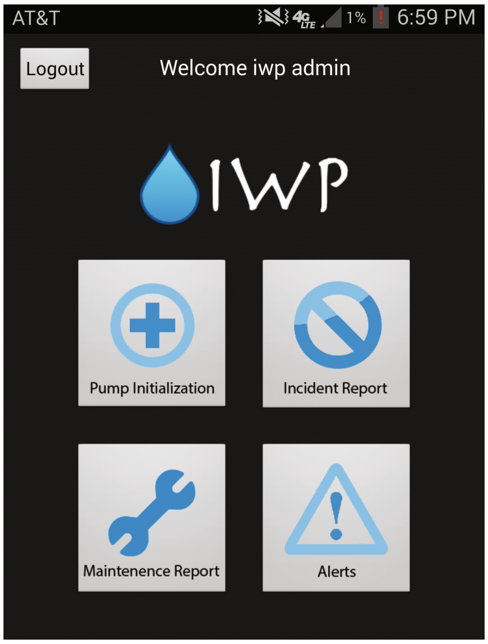

Fig. 5. IWP mobile app.

Alerts: This function allows a user to view the alerts (as previously described) associated with the handpumps for which the user has been granted access.

Offline and Synchronization Modes: The mobile app requires a data grade connection to transmit data, and since there will be times when such a connection is not available, an offline and synchronization mode is necessary. In such cases, the mobile app will locally persist the data (i.e. yet to be filed maintenance reports and incident reports) on the mobile device. Upon the mobile app sensing a data grade connection, the persisted data will be transmitted to the IWP system.

There are several important future considerations for the IWP mobile app. The current mobile app is purpose-built for the Android Operating System® and uses the Android SDK®. Given the trade-offs of platform-specific mobile apps versus responsive web app design, we would opt for a responsive web design in the future that supports a myriad of mobile device types.

Secondly, we would incorporate location awareness features in the app that would allow a user to check for pumps near them or to be given directions to a specific pump. We would also extend the app to include multi-language support. Finally, we believe pumps should become part of the "Internet of Things" ("IoT") economy and themselves become network addressable using NFC, thereby allowing authorized users to obtain data about a pump by using an NFC-enabled mobile app.

5 Summary and Future Work

A key purpose of this paper was to outline the key software design decisions made for IWP. The real-world environmental factors for deploying IWP (climate, lack of reliable Internet access, solar power, etc.) all made for a set of interesting design constraints. Hopefully, this paper provided some insight into the key software design decisions made and rationale for the same against these imposing design constraints.

Significant opportunities remain for advancing our work. Sensor data collected over time will validate assumptions and features such as alerts, change in handpump status, etc. Ideally, we would like to see the system deployed through the status cycle of a handpump to insure that the system correctly senses the deterioration of the handpump, issues the appropriate alerts and senses the handpump performance improving upon being repaired by a field technician.

In addition to the future directions of IWP outlined in the paper, the IWP software will continue to evolve based on lessons learned from field trials. It is expected that significant advances will be made in handpump data analytics based on feedback from handpump field technicians. Beyond the ideas mentioned in the paper, there are a number of future directions envisioned for IWP, including:

1. publishing and supporting APIs (data import, data export, big data analysis, etc.) and a platform for third party application developers to integrate with, innovate on and extend our platform and the capabilities provided by IWP,
2. creating a collaboration capability for researchers, field water technicans, community leaders, NGO staff and funders to be able to dialog and share information about pump performance and related data,
3. creating anonymous data aggregation and analyses capabilities for organizations and other interested parties to compare their pump performance and related data to such data from other pump groups.

Acknowledgements. We acknowledge our hardware counterparts, Dr. David Vader and Anthony Beers along with various Computer Science and Collaboratory students for their work on this project.

References

1. RWSN: Handpump data 2009 (2009)
2. Nejmeh, B., Weaver, D.S.: Leveraging scrum principles in collaborative, interdisciplinary service-learning project courses. In: Frontiers in Education Conference (FIE), pp. 1–6. IEEE (2014)
3. Weaver, D.S., Nejmeh, B., Vader, D., Beers, T.: The intelligent water project: bringing understanding to water pumps in Africa. In: Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM), vol. 1, pp. 211–218 (2016)
4. Beers, A.Q., Vader, D.T.: India mkii pump sustainability study report. Technical report, Messiah College (2013)

5. MacArthur, J.: Handpump standardisation in Sub-Saharan Africa (2015)
6. Sansom, K., Koestler, L.: African handpump market mapping study (2009)
7. Crockford, D.: The application/JSON media type for Javascript object notation (JSON)(2006). http://tools.ietf.org/html/rfc4627
8. Nejmeh, B.A., Dean, T.: The charms application suite: a community-based mobile data collection and alerting environment for HIV/AIDS orphan and vulnerable children in Zambia. Int. J. Comput. ICT Res. **46** (2010)
9. Upside Wireless (2016). http://www.upsidewireless.com/
10. New Digital Group, Inc.: Smarty template engine (2016). http://www.smarty.net/
11. Google: Google maps APIs (2016). https://developers.google.com/maps/documentation/javascript/
12. esri.com: About Esri (2016). http://www.esri.com/about-esri#who-we-are