6-11-2015

iKure health worker tracker

Daniel Mendoza
Santa Clara University

Astha Singh
Santa Clara University

Mason Maeshiro
Santa Clara University

SANTA CLARA UNIVERSITY

Department of Computer
Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Daniel Mendoza, Astha Singh,
Mason Maeshiro

ENTITLED

iKure Health Worker Tracker

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING
BACHELOR OF SCIENCE IN WEB DESIGN & ENGINEERING

Thesis Advisor

Department Chair

iKure Health Worker Tracker

By

Daniel Mendoza, Astha Singh,
Mason Maeshiro

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Computer Science &
Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements for
the degree of

Bachelor of Science in Computer Science & Engineering
Bachelor of Science in Web Design & Engineering

Santa Clara, California

2014-2015

Table of Contents

List of Figures

Abstract

For our Senior Design Project, we worked with a company in India called iKure. This company is a social benefits entrepreneurship that strives to provide affordable healthcare to citizens in rural India. In order to run their operations more efficiently, iKure asked us to create an application to track the location of their health workers as they travel throughout the rural communities. We worked closely with one of the employees at iKure, meeting bi-weekly through conference calls, in order to keep up to date on the company's developments.

After careful consideration and additional feedback from iKure, we came up with a two-part system. The first is a mobile application developed using Phonegap that will be installed on the android tablets used by the health workers. This application automatically sends the worker's location data to our database in specified time intervals during the designated tracking period. The second part of our system is a website, accessible only by the iKure administrators, which gives them the ability to view the current location of all health workers as well as any worker's locations in the past week.

The scope of our health worker tracker is not only limited to iKure. It has the potential to be expanded to other industries and services, like emergency response services. By utilizing our product, companies and supervisors will be able to monitor their workers in the field. Because they will know the general location of each worker, supervisors can quickly and efficiently send their workers to any situations that arise.

Chapter 1 - Introduction

1.1 Problem Statement

iKure addresses the pressing need of healthcare in India's rural area by providing accessible and affordable primary care through the use of technology. Health workers form a bridge between doctors and patients. Through Android tablets, healthcare workers in the field are able to send data and reports to professional doctors across the country. Health workers carry these tablets while traveling to rural communities surveying each individual house, while marketing iKure. iKure keeps track of the hours worked by the health workers manually with a paper check-in. A manual paper to record hours is error-prone and inefficient. A validation method is required.

Currently, local clinic supervisors have no way to oversee the day-to-day actions of their own subordinates out in the field. Despite iKure's aspirations to provide health care services to rural communities through the use of cutting technology, the company's methodology for monitoring its own employees is quite limited. Once a week, the health workers fill out generic paper reports to record the number of hours they worked in the community, specific places they visited, and data they collected over the past 7 days. The major flaw in the current system is that it does not contain a method to hold employees accountable. At the moment, health worker hours and routes cannot be recorded thus forcing supervisors to accept any data reported as fact. Therefore, the entire legitimacy and accuracy of the weekly report relies completely on the memory and morality of the health worker. The non-existent process for tracking and checking the health worker's path greatly increases the probability and temptation of false reporting, which hinders iKure's ability to operate effectively and further decreases the potential benefits of iKure health care services across the country. Additionally, the current arrangement lacks a real-time monitoring system required to efficiently manage the numerous mobile workers constantly traveling throughout the country's rural communities. Health workers must follow set schedules with the goal of visiting each site twice a week.

In order to solve this issue, we designed a simple, user-friendly system to easily track, store, and view the location data from each health worker. The development process of our solution involved two parts. The first part of the development was devoted to creating a mobile application that automatically starts and stops tracking the location of health workers at specific times and stops, sending the worker's location in set time intervals. For each location update, the current date and time is also stored allowing that data to be easily sorted and viewed by the administrator. We built the mobile application using PhoneGap, a mobile development framework that converts code from modern web programming languages (HTML, CSS, Javascript) into a fully functional mobile application that can be installed on practically any type

of phone or tablet. The second part of the design included creating a website and database to store and view the location data collected from the health workers. The website is only accessible to iKure administrators and allows them to easily view the current or past locations of individual health workers. This allows iKure's supervisors more command and control over the workers, increasing efficiency and accountability.

1.2 Requirements

Requirements Overview

In order to address the problem discussed in Section 1, requirements and design constraints must be considered when implementing the proposed solution. The two types of requirements that the application must adhere to are functional and nonfunctional requirements. Functional requirements define what the program must accomplish. Non-functional requirements define the manner in which the functional requirements need to be completed. The project is also subject to some design constraints. Design constraints affect the way the solution is designed and implemented. The specific requirements and design constraints are discussed below.

Functional

The functional requirements for the Android application are as follows:

- The application should track the user's location in set time intervals
- The application should automatically start and stop tracking at designated times
- The application should work both with 3G or Wifi signal
- Administrators should have accounts to access health worker location data
- The application should have a search feature in which the administrator can look up a search worker currently working in the community and see his or her location.
- The past location data should automatically be removed from the database after one week

Non-Functional

The non-functional requirements for the Android application are as follows:

- The program should be usable by people with various technological backgrounds.
- The application should run on an Android tablet.
- The application should store the location data in a protected database

Design Constraints

The design constraints for the Android application are as follows:

- The mobile application should work on Android OS
- The website should function on all browser types
- Both the application and website should work outside of the United States (India)

1.3 Societal Issues & Implications

The primary issue or concern with building our product was the issue of the workers' privacy. However this concern was alleviated by adding start and stop times to the location tracking algorithm. This allows the administrator to tailor the application to the specific needs of that worker, building the tracking settings around his or her work shift. Additionally, iKure only intends on installing our mobile application on the company-owned android tablets that it lends out to the health workers. Since the tablets are owned by iKure, it's acceptable to track the location of its own property.

Our product is not only limited to the scope of iKure's health worker network. It can be expanded to other industries and services, like emergency response services for example. The location of emergency responders could be constantly tracked so in the event of a natural disaster or other serious situation, supervisors could find and direct the closest responders to the scene. This would allow for quicker response times, increasing the chances of survival for anyone involved in the disaster.

Chapter 2 - System

2.1 Overview

Our system consists of a mobile application, website, and database. The database stores the location information being sent from the mobile applications installed on the health workers' android tablets. Only administrators are able to log into the website and view both the workers' current locations as well as past locations, sorted by day. Through the website, administrators also have the ability to create new administrator accounts. After one week, the location data stored in the database will be automatically deleted, allowing the database to be managed without any administrator assistance.

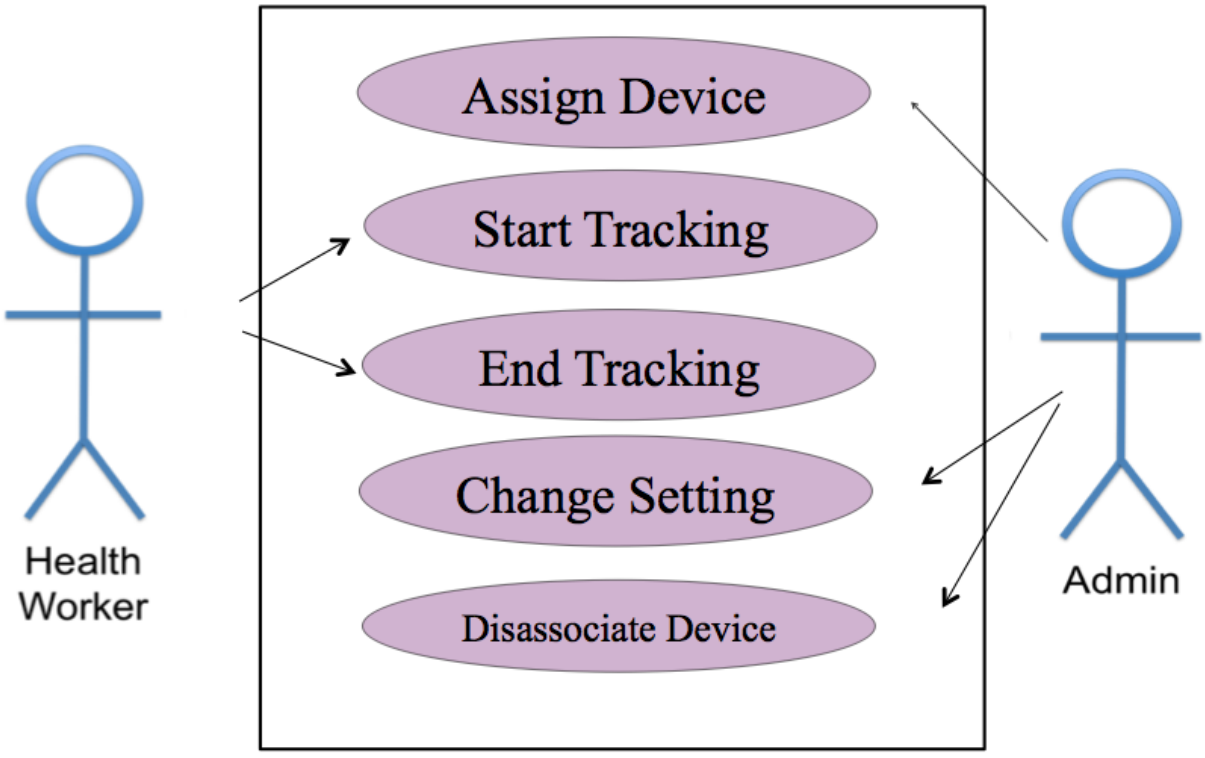
2.2 Use Cases

After the conceptualization of the model, an analysis of how the user will interact with the web application must be made. We have created a set of use cases, or actions taken by the user, in order to demonstrate the different scenarios that a user may find himself in, as shown in the conceptual model figures. Our mobile application will have two independent sets of features, one for health workers and one for administrators who supervise those health workers.

A use case typically includes:

- Actors
- Goal
- Precondition: Things that must be true before the use case can be performed
- Postcondition: Things that will be true after the use case has been successfully performed
- Steps Required: Enumerated list of steps, sometimes w/ substeps or alternatives
- Exceptions: Common errors and how they are resolved

Figure 1: Mobile Application Use Case Diagram - Available actions on the mobile application



Mobile Application Use Cases:

- Assign Device
    - Actor: Administrator
    - Goal: Administrator successfully inputs health worker information to begin location tracking
    - Preconditions:
        - Mobile application downloaded successfully on the device
    - Postconditions:
        - Health worker sees the home screen
        - Location tracking is initialized
    - Steps Required:
        1. Actor accesses mobile application (See Figure 3)
        2. Actor inputs the necessary information, changing any settings if necessary
        3. Actor presses the submit button
    - Exceptions:
        - The device is not connected to the internet

- Start Tracking
    - Actor: Health Worker
    - Goal: Location tracking has started
    - Preconditions:
        - Device is assigned to a health worker (see Figure 4)
        - Location tracking start time has passed
    - Postconditions:
        - Location data is stored in the database
    - Steps Required:
        1. (none) - Location tracking starts automatically
    - Exceptions:
        - Device not connected to the internet

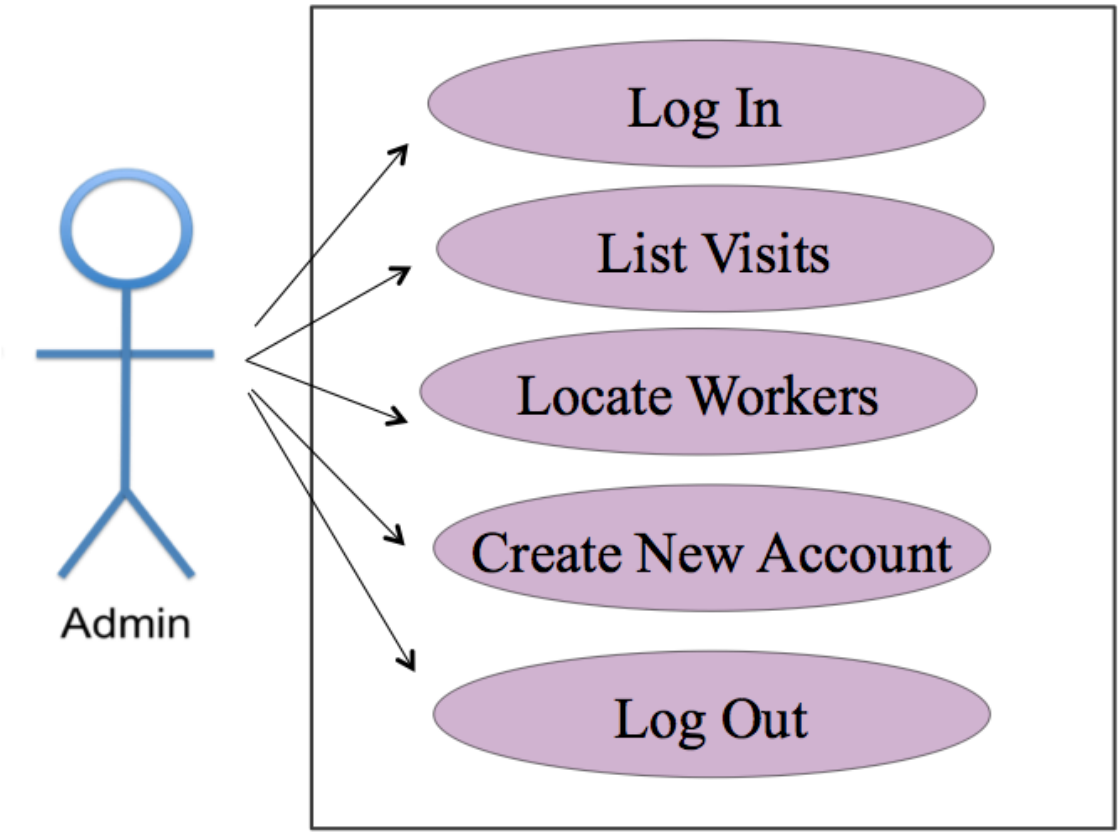
- End Tracking
    - Actor: Health Worker
    - Goal: Location tracking has stopped
    - Preconditions:
        - Device is assigned to a health worker
        - Location tracking has started
        - Location tracking stop time has passed

    - Postconditions:
        - Location tracking has stopped
    - Steps Required:
        1. (none) - Location tracking ends automatically
    - Exceptions:
        - None

- Change Settings
    - Actor: Admin
    - Goal: Location tracking settings have been updated
    - Preconditions:
        - Device is assigned to a health worker
        - Administrator has a valid admin account
    - Postconditions:
        - Location tracking settings have been saved to the device
    - Steps Required:
        1. Actor clicks on the Admin Settings button (see Figure 5)
        2. Actor inputs valid admin credentials and clicks submit
        3. Actor changes any necessary information (see Figure 6)
        4. Actor clicks the Save and Exit button
    - Exceptions:
        - Device not connected to the internet

- Disassociate Device
    - Actor: Administrator
    - Goal: Administrator clears device of health worker information
    - Preconditions
        - Administrator input valid credentials
        - The application is assigned to a health worker
    - Postconditions:
        - Location tracking has stopped
        - Health worker information is cleared from the device
        - Application now displays the assign device screen (see Figure 3)
    - Steps Required:
        1. Administrator logs into the Admin Settings
        2. Administrator clicks the Reassign Device button (see Figure 7)
        3. Administrator clicks the confirmation button
    - Exceptions
        - None

Figure 2: Website Use Case Diagram - Available actions to the administrator on the website



Website Use Cases:

- Log in
  - Actor: Admin
  - Goal: Admin successfully logs into his or her account
  - Preconditions:
    - Website loaded onto browser
    - Actor has an admin account
  - Postconditions:
    - Admin has the options to locate health workers, list health workers, create an account, and log out (see Figure 9)
  - Steps Required:
    1. Actor loads website url, loading the login page (see Figure 8)
    2. Actor inputs his or her username and password correctly
    3. Actor presses the submit button
  - Exceptions:
    - The actor inputs the wrong url

        - The actor inputs incorrect username or password
        - The device is not connected to the internet

- List Health Workers
    - Actor: Admin
    - Goal: View past visits of the selected health worker
    - Preconditions:
        - The actor is logged into an admin account
        - The device is connected to the internet
    - Postconditions:
        - The selected health worker's information and past visits are displayed
    - Steps Required:
        1. The actor clicks "List Health Workers" button
        2. The actor chooses one of the listed health workers (see Figure 10)
    - Exceptions:
        - The device is not connected to the internet
        - No workers are currently assigned to a device

- Locate Health Workers
    - Actor: Admin
    - Goal: View current location of different health workers
    - Preconditions
        - Actor is logged into an admin account
        - The device is connected to the internet
    - Postconditions
        - The selected health worker's current location is displayed on the map (see Figure 13)
    - Steps Required:
        1. Actor clicks "Locate Health Workers" button
        2. Actor chooses a health worker from the list of currently logged in workers
    - Exceptions:
        - The device is not connected to the internet
        - There are no health workers currently assigned to devices

- Create New Account
    - Actor: Admin
    - Goal: Create a new health worker account
    - Preconditions:
        - The device is connected to the internet

        - An account does not exist for that administrator
    - Postconditions:
        - An admin account has been created from the inputted information
        - The account's generated credentials are stored in the database
    - Steps Required:
        1. Actor selects "create new account" button
        2. Actor inputs all required information
        3. Actor presses submit button, loading the status message (see Figure 17)
    - Exceptions:
        - The device is not connected to the internet
        - The actor did not fill in all required information (see Figure 15)
        - The password and re-entered password do not match (see Figure 16)
        - An account for that administrator already exists

- Log Out
    - Actor: Admin
    - Goal: Admin logs out of his or her account on the website
    - Preconditions
        - Actor is successfully logged into his or her account
    - Postconditions:
        - The actor is logged out of his or her account on the website
    - Steps Required:
        1. Actor clicks on "log out" button
    - Exceptions
        - None

2.3 Changes

Throughout the development process, our system changed and evolved from the original conceptual model. Working with a client introduces its own set of benefits and challenges. We were able to constantly receive feedback from our client to help ensure our product was fulfilling their needs. However, one challenge we encountered was that our client kept introducing new requirements as time went on. As engineers, it was our job to prioritize these requirements and ultimately decide what we could implement and what we couldn't with the time remaining.

Our first major changes occurred after we submitted our design document to our client. Originally we planned on creating one product which contained both the health worker and administrator features. Through Phonegap it was possible to create both a web application as well as a mobile application of the same code and based on the login credentials provided, the application would grant the user access to either the health worker features or the admin features. However, this all changed when our client told us that they did not want health workers to have accounts stored on the database. They also didn't want health workers to be able to see the google map of their location nor their past visits. They also stated that the administrators would only need to view the administrator features on a computer, so we ultimately decided to create two independent products - a mobile application and a website.

Our next big change came when we submitted the demo video of our initial operational system to our client. Originally our client told us to start tracking at 8:00am and stop tracking at 6:00pm, however now they asked us to allow administrators the ability to change those settings on the application. This brought the admin settings feature to the mobile application.

About 3 weeks before the Senior Design Conference, our client introduced several other big additions. First, they wanted the administrator to be able to search for a specific health worker on the website. Secondly, they wanted to implement a "zones" feature, which allowed an administrator to divide the area into different segments and sort the health workers based on which segment they were located in. With so little time left, we ultimately decided that we were unable to add the zones feature, but still successfully implemented the search feature to our website.

2.4 Final Product

The following figures depict the different pages displayed on our final product. Figures 3 - 7 are from the mobile application, while Figures 8 - 17 are from the website.

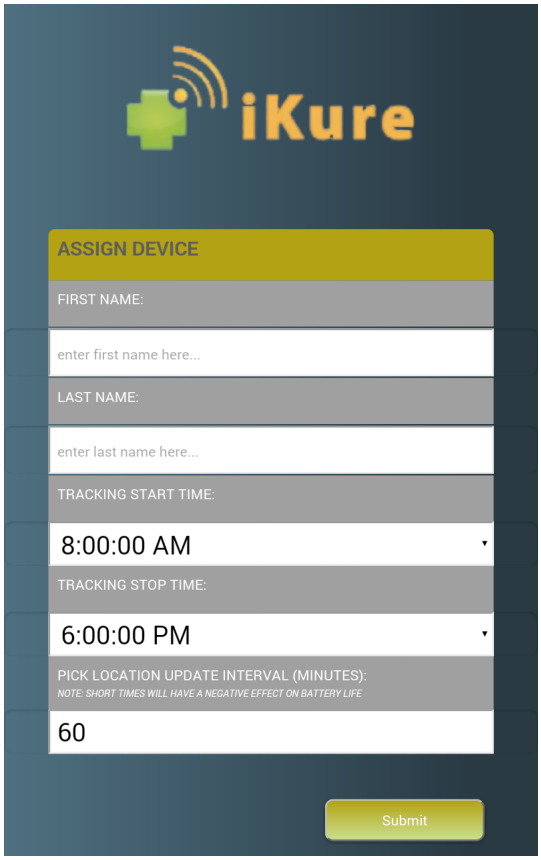


Figure 3: Assign Device Page - Admin assigns the device to a health worker

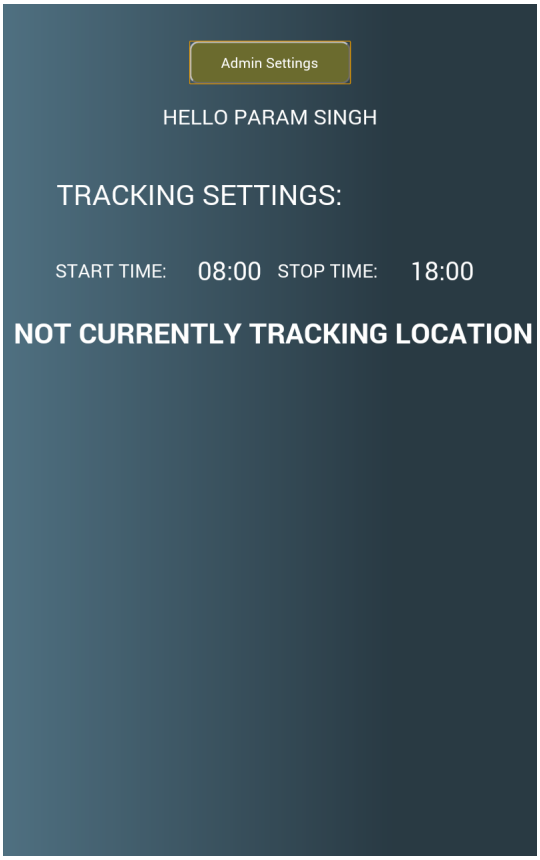


Figure 4: Home Page - Displayed once device is assigned to health worker

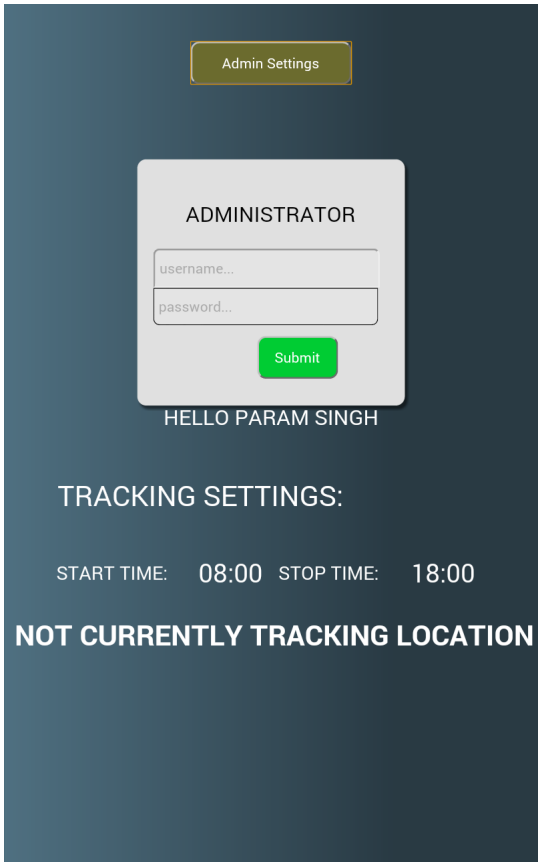


Figure 5: Admin Settings Login - Displayed to allow admin access to settings screen

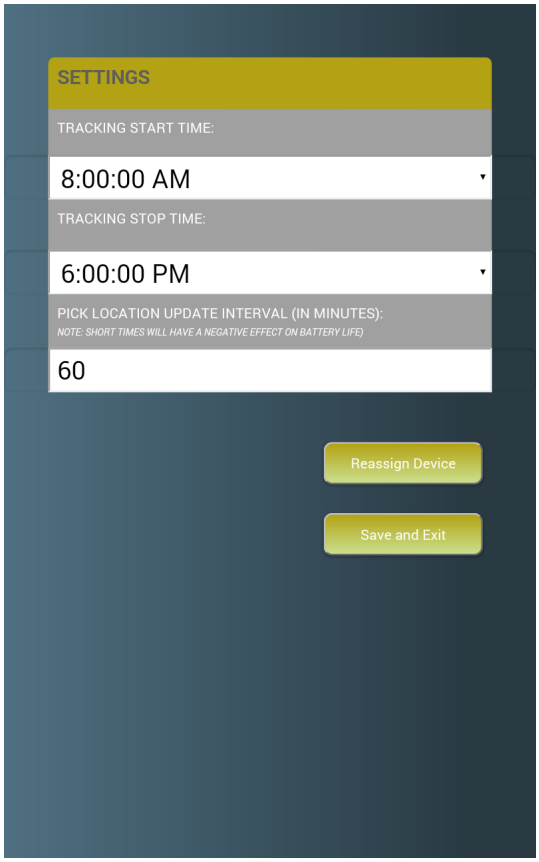


Figure 6: Admin Settings Page - Allows admin to change the tracking settings

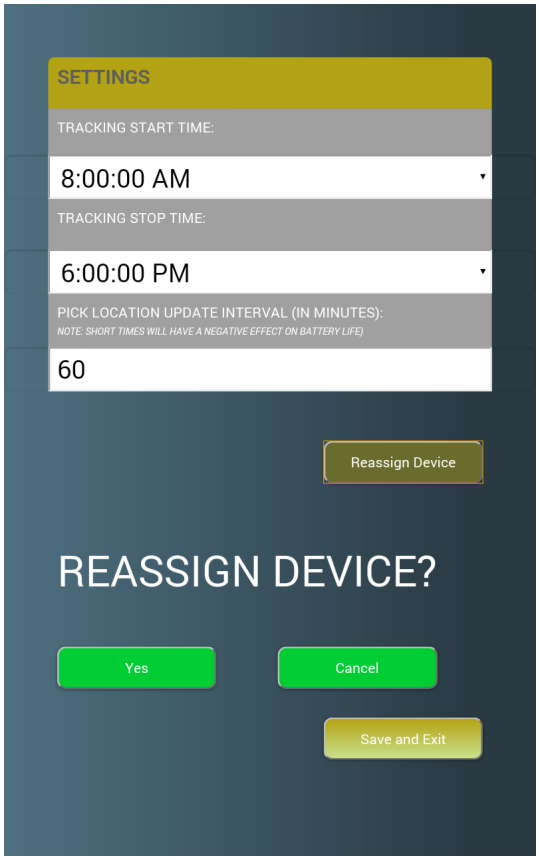


Figure 7: Reassign Device - Clears all assigned worker information from device

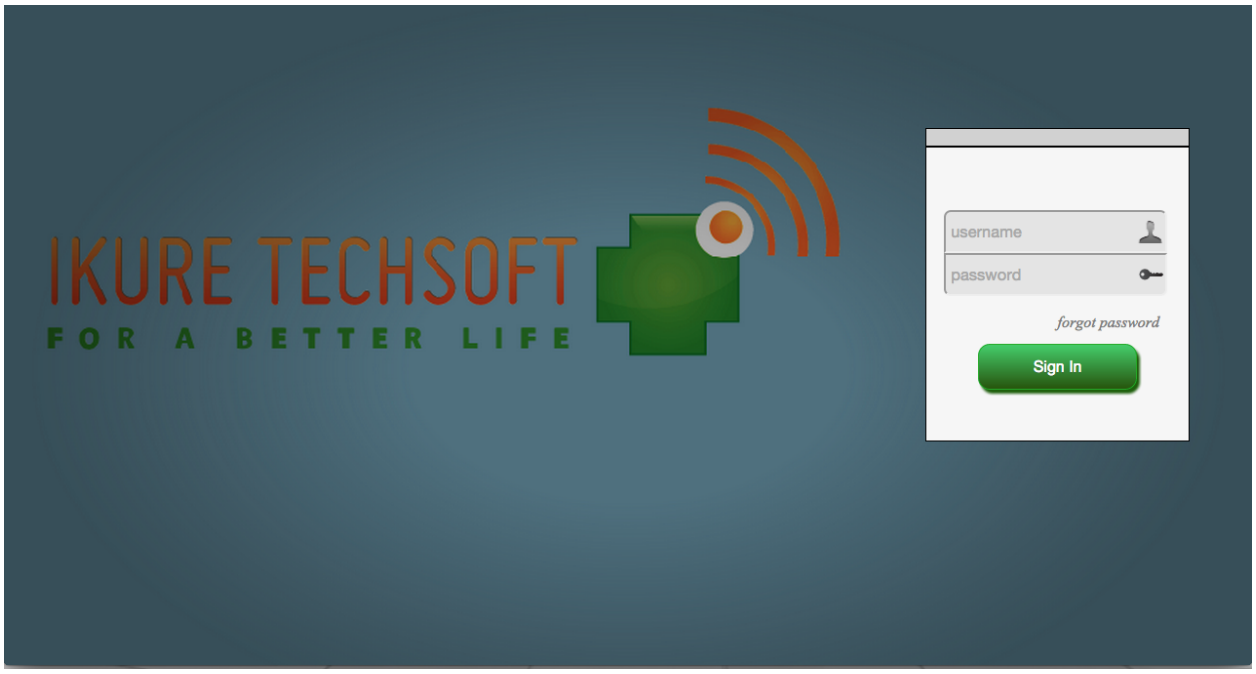


Figure 8: Login Page - Only allows administrators to access to website

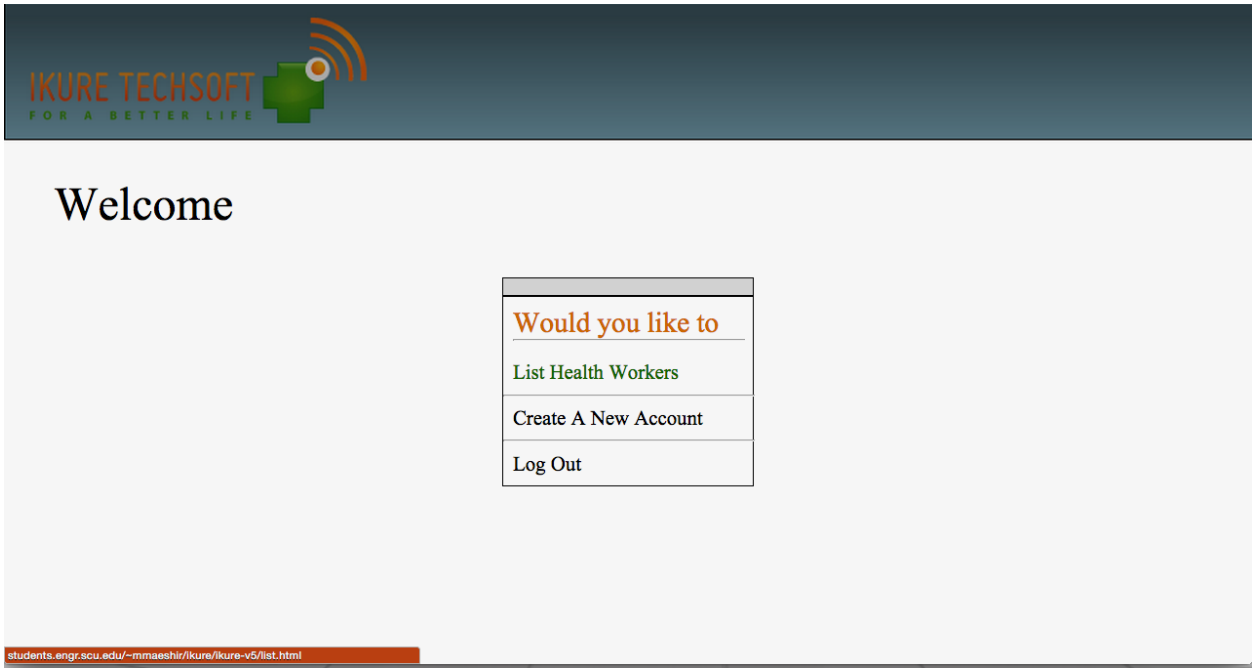


Figure 9: Welcome Page - Lists the different options available on the website

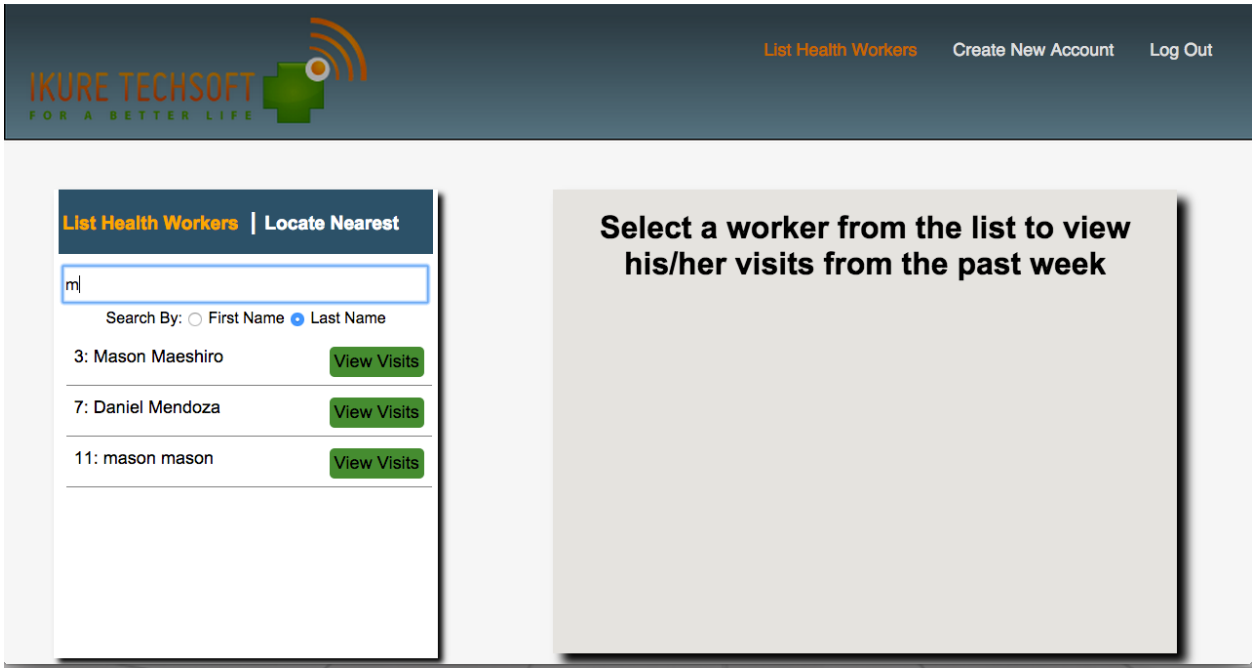


Figure 10: List Health Workers Page - Lists all health workers currently assigned to devices (currently displaying search results)

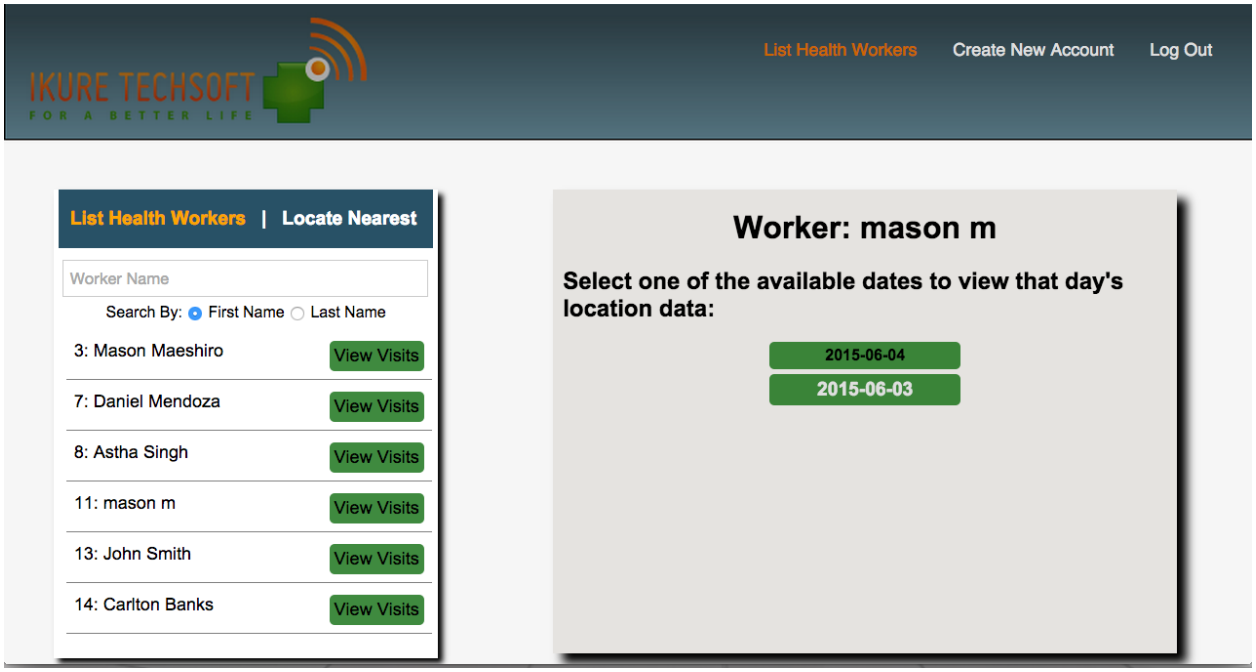


Figure 11: View Worker Visits - Lists the worker's visits from the past week

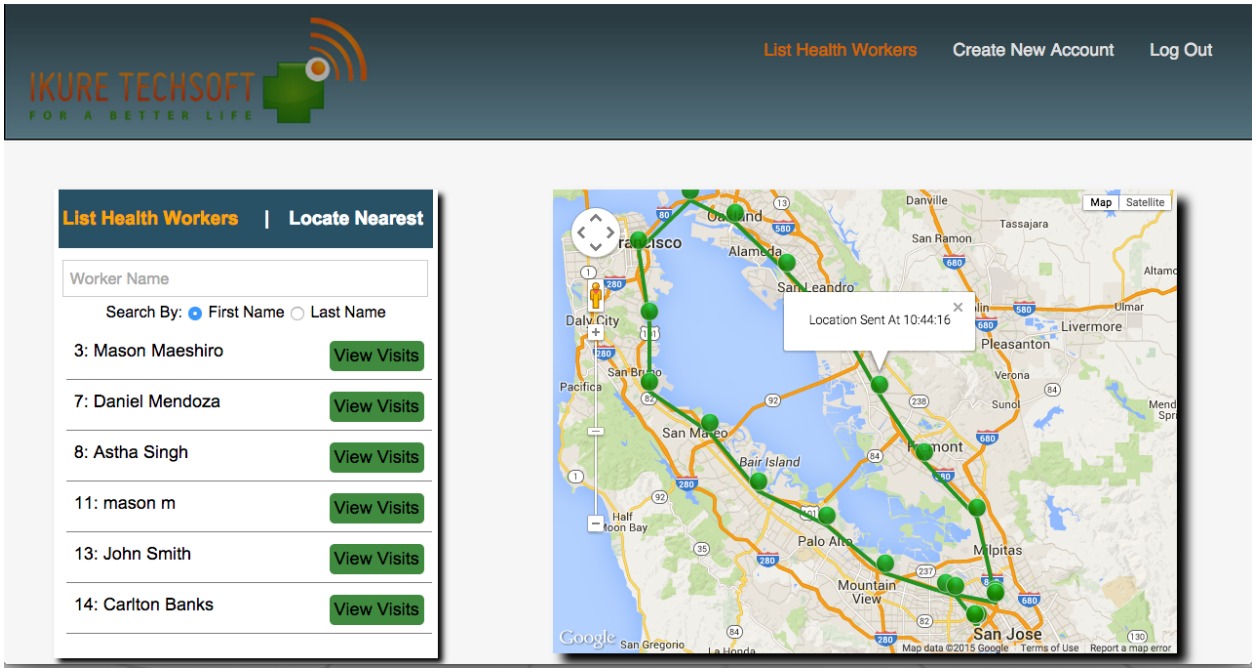


Figure 12: View Visit - Plots the location data from the specific visit on the map

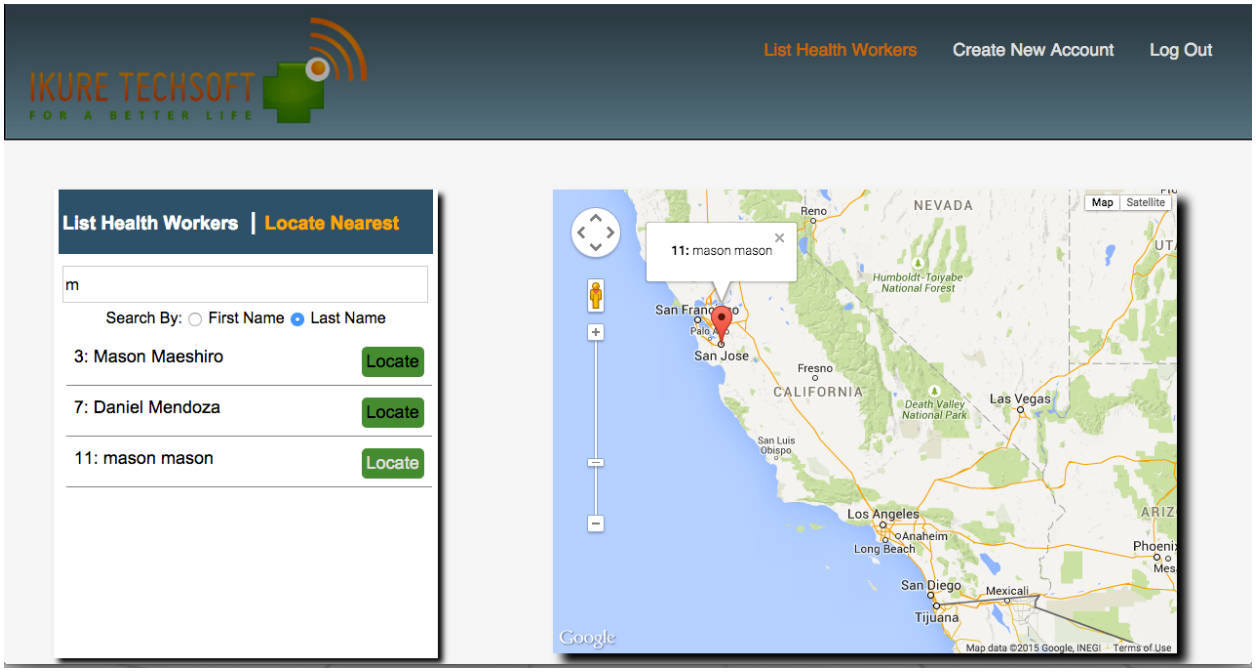


Figure 13: Locate Nearest Worker Page - Plots worker's last known location on the map

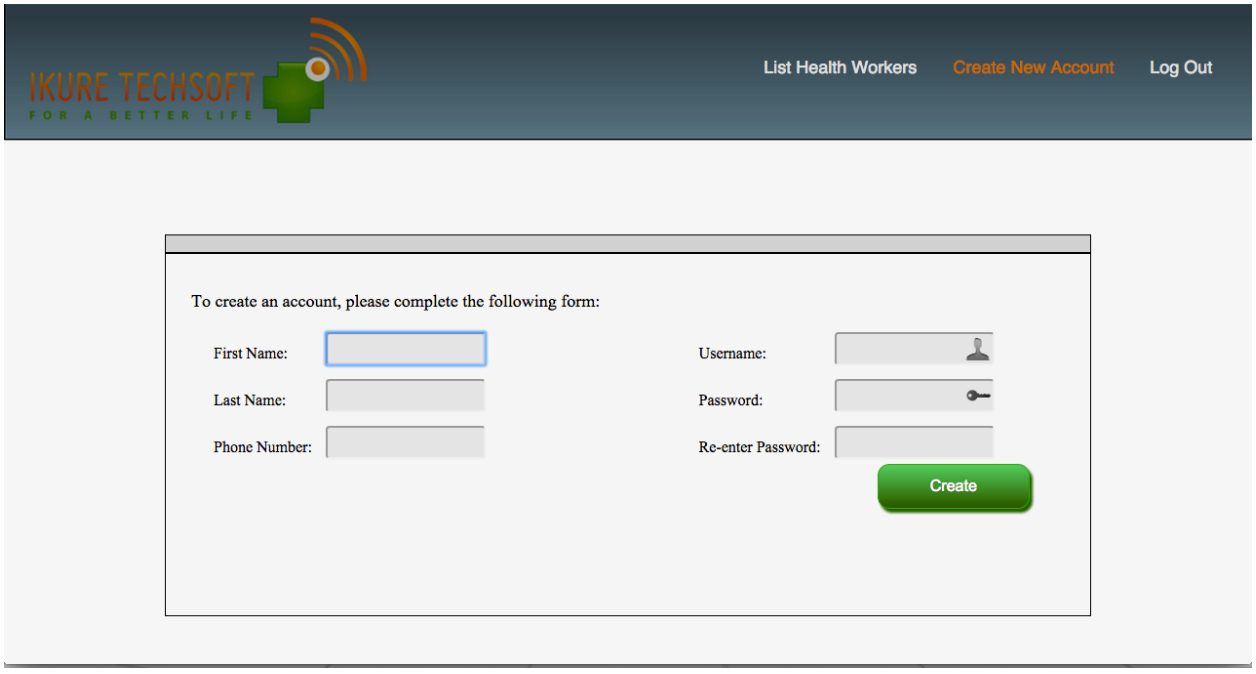


Figure 14: Create New Account Page - Allows admin to create another admin account

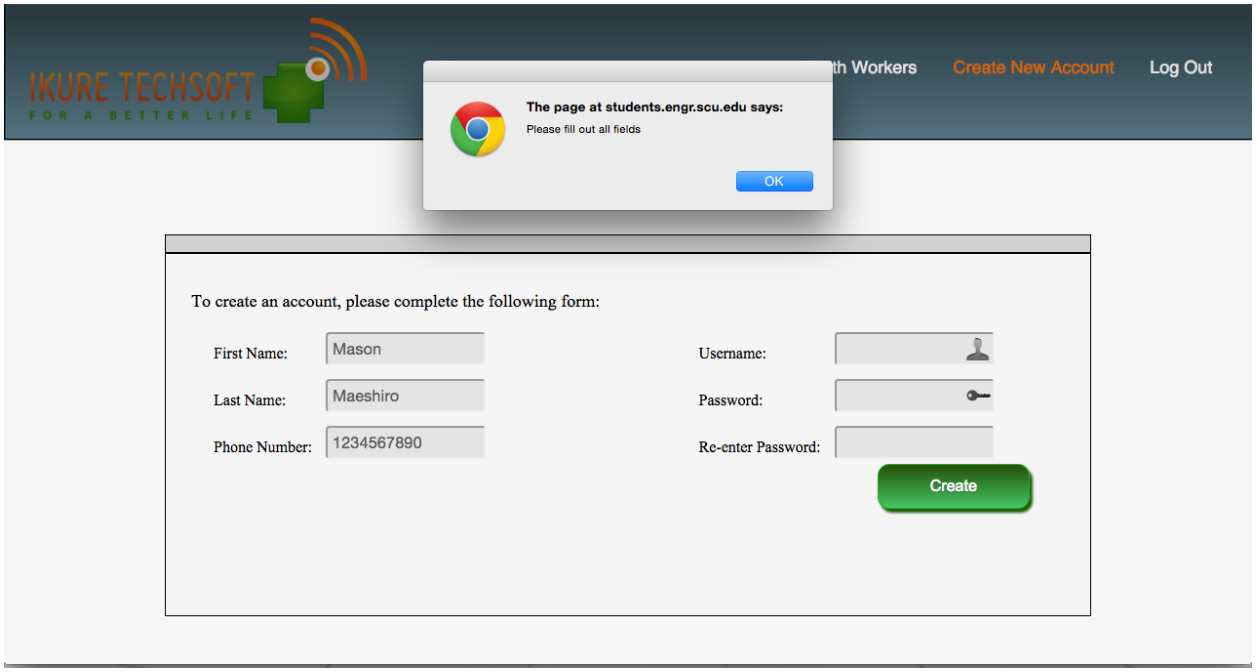


Figure 15: Missing Information Alert - Prevents creating an account with missing information

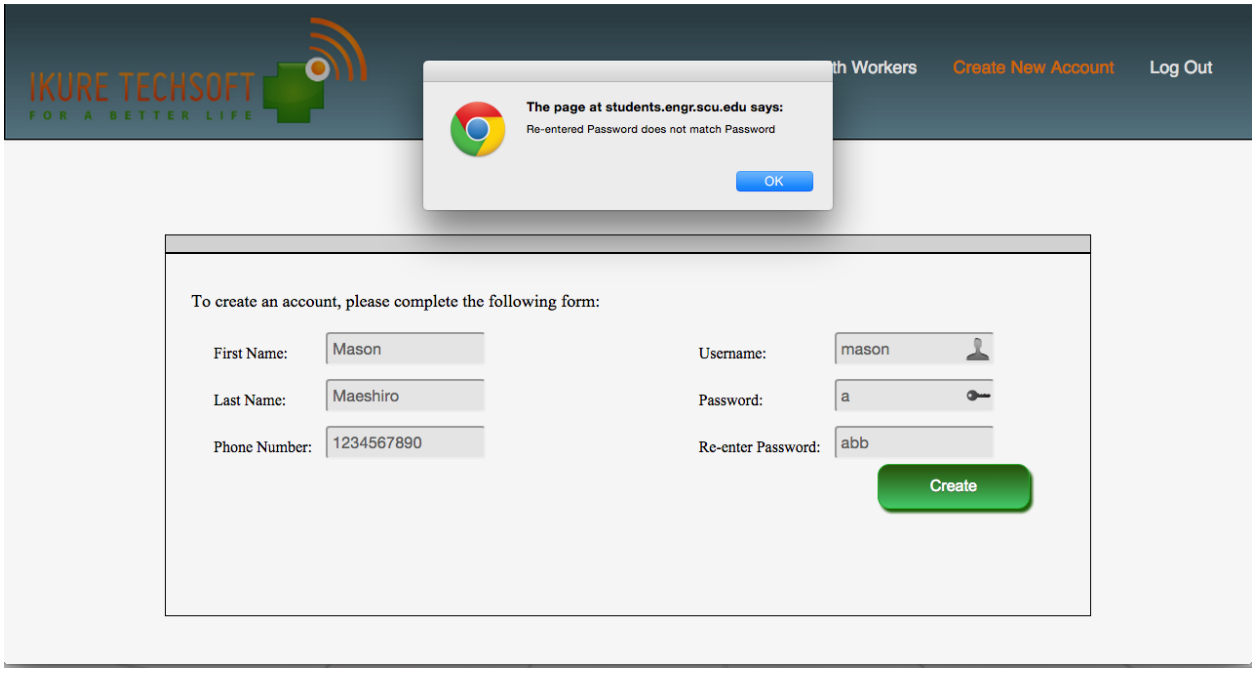


Figure 16: Password Mismatch Alert - Ensures the password matches the re-entered password

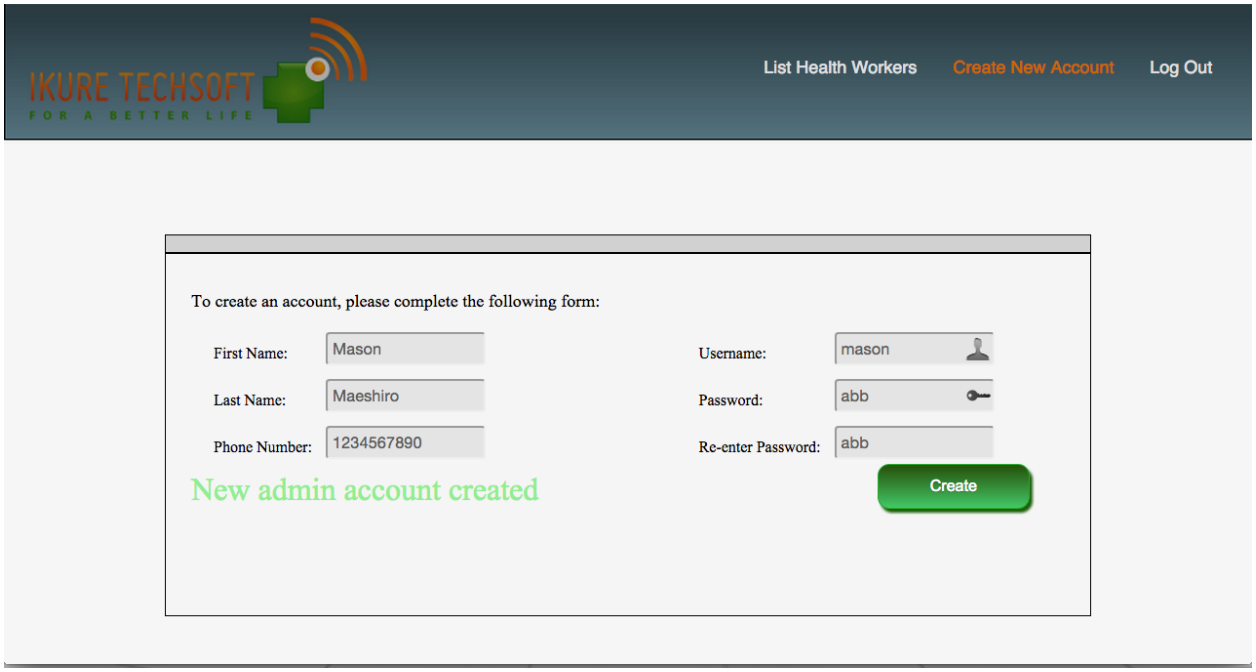


Figure 17: Create Account Success - Success message if account has been created

2.5 Technologies Used

Our system was created using many different technologies. We needed to create a database, mobile application, and website, introducing many different functionalities and features that we needed to implement. The following technologies will be used:

- HTML - HyperText Markup Language. HTML forms the foundation for the web page, and contains the code that references the other components like Javascript and CSS files. The HTML acts like a wrapper the entire system, bringing it all together.
- CSS - Cascading Style Sheets. These contain the style that is used in the webpage, defining things like margins, colors, borders and banners.
- JavaScript - Javascript is a dynamic programming language commonly used for websites and online applications.
- jQuery - Jquery is a Javascript library that allows simple and seamless integration of animation into static pages or applications. For the most part, we will be utilizing the jQuery mobile library to program our application.
- Phonegap - Phonegap is a mobile development framework that allows us to use simple web programming languages (HTML, CSS, JavaScript, Jquery) and automatically converts those program files into a readable, executable application.

- PHP - PHP is a server-side scripting language designed for web development that allows us to communicate with our remote database.
- MySQL - MySQL is a database language that allows for simple creation of tables to store all types of information for later use.

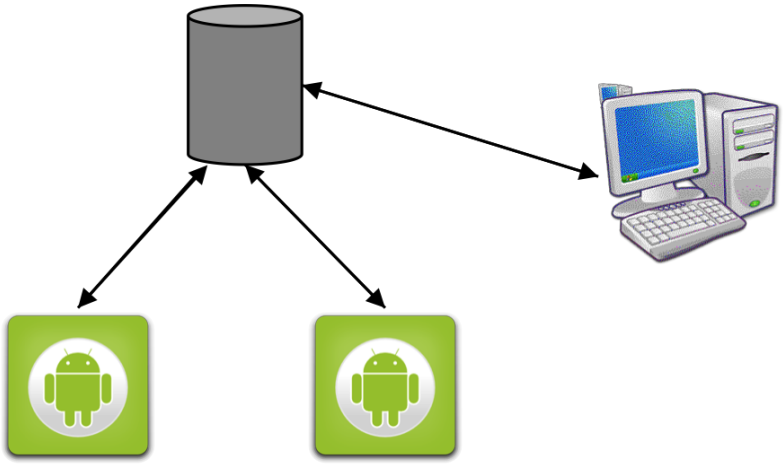
2.6 Design Rationale

We decided to utilize Phonegap to develop our mobile application. This framework offers cross-platform compatibility, which allows the application to be downloaded to different devices if our client chooses to expands beyond Android tablets in the future. Applications created with the Phonegap framework are developed with simple web programming languages: HTML, CSS, Javascript, and jQuery. HTML is used to define the basic structure of the web pages, while CSS applies the custom styling to those web pages. Javascript is the programming language that runs all logic and functionalities of our application, while jQuery provides user-friendly animations to increase visual appeal and usability. We used the same web programming technologies to develop the website side of our system.

We used a MySQL database to store the location and user account information because it is easily scalable and can easily handle search queries for large amounts of data. This allows our application the ability to quickly pull or store the location data of any health worker for both the mobile application and website sides of our system. In order to communicate with the database, we are using PHP scripts, which are hosted on a web server, to define the necessary MySQL queries for the different features of our system.

As shown in Figure 18, each application will communicate solely with our MySQL database through the PHP scripts. The PHP scripts automate all inserting, searching, and deleting, decreasing the need to manually maintain the large amount of data coming from our mobile application and website. Each application and website instance will operate independently of all other instances, simplifying our design and implementation.

Figure 18: System Architectural Diagram



Chapter 3 - Project Management

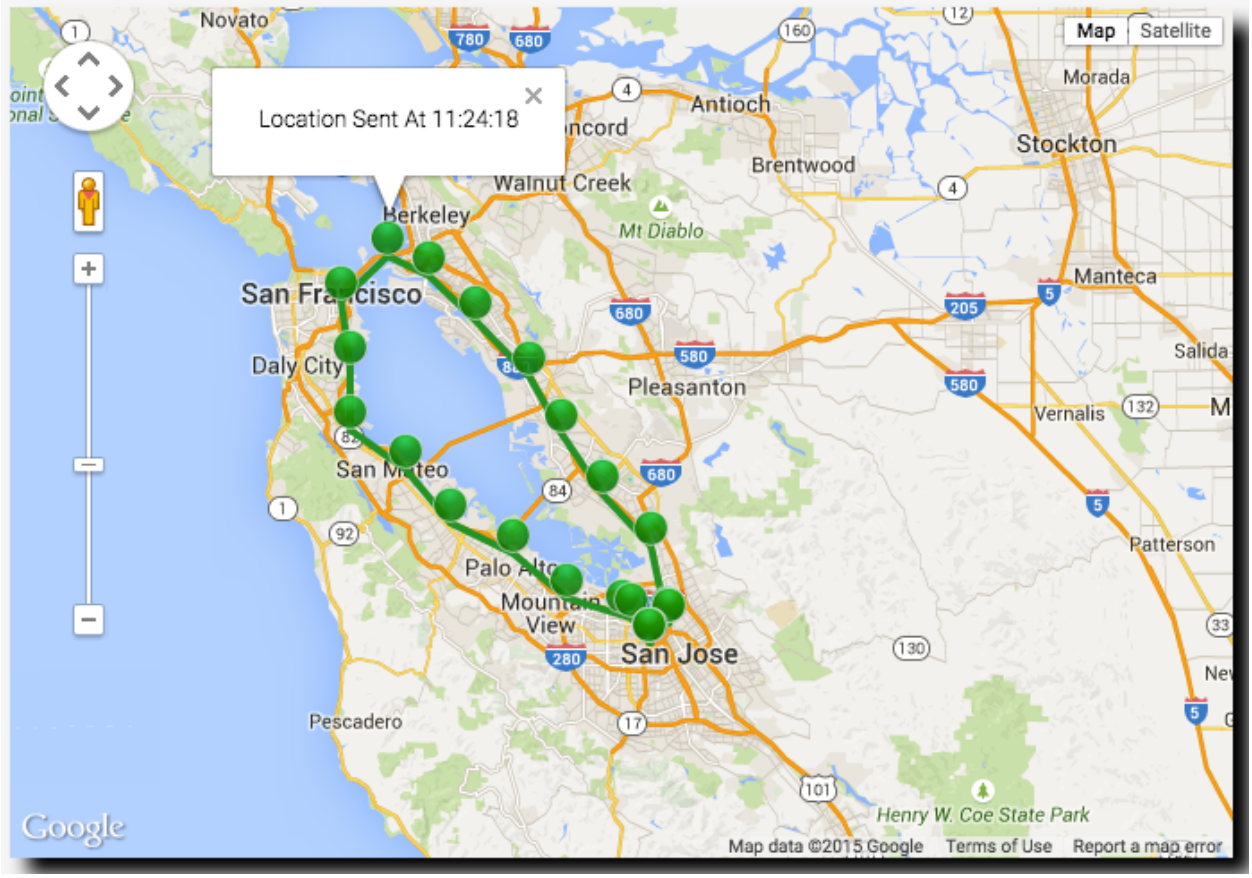
3.1 Project Risks

Risks	Consequences	Probability	Severity	Impact	Mitigation Strategy
Bugs	Not being able to work properly	1.0	5	5.0	Thoroughly test all cases as we develop the program. Have classmates beta test our system to discover unknown bugs.
Requirements Clarity	Not being able to meet requirements	0.7	7	4.9	Meet with the customer on a regular basis.
Group Management	Not being able to get together for meetings due to conflicting schedules or illness	0.5	5	2.5	Make sure we communicate with one another. Making sure we are all able to understand each subsection. Have co-leads for each section.
Customer Adding Requirements	Not being able to coordinate and finish on time	0.1	8	0.8	Work on sections periodically and being prepared each week for any extra requirements. Working ahead of schedule just in case we run into issues

3.2 Test Plan

Throughout the implementation process, various tests have been conducted to ensure our application runs smoothly and successfully. We utilized both white and black box testing. An Android tablet has been used to test our application periodically. We tested by choosing different scenarios for the health worker. Unit testing has been used on our module to ensure the validity of information and schedule of health workers. Additionally, as seen in Figure 19, we simulated a drive by a health worker and modified the update interval to 10 minutes to test the view past visits feature of our website. Customer acceptance testing will be done by our customer to ensure our application works correctly during the summer of 2015. Through the Global fellows Program at Santa Clara University, two Santa Clara University students will be travelling to India to over the Summer of 2015 to deploy this application and fix potential bugs that may arise with the deployment of this application in another nation.

Figure 19: Route Simulation - Map drawn from simulated route data of a trip around the San Francisco Bay



3.3 Development Timeline

The development timeline for this application covered an entire school year, starting from September 22nd and ending on May 8th. Figure 20 depicts the timeline breakdown by each quarter, indicating the task assignments, amount of time for the task, and the team member who was assigned to complete the task. This allowed our team to stay on schedule throughout the production process.

Figure 20: Application Development Timeline

TASKS	Fall Quarter (9/21 - 11/1)					
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
	9/21 - 9/27	9/28 - 10/5	10/6 - 10/11	10/12 - 10/18	10/19 - 10/25	10/26 - 11/1
Problem Statement			X			
- Scenarios					X	
- Technical Questions						
Research						
- SMS application						
- Android application						
- Phone Gap						
- Database/Server						
- Location tracking methods						
Design Document						
- Title Page/Table of Cont.						
- Requirements						
- Conceptual Model						
- Use Cases						
- Architectural Diagram						
- Technologies Used						
- Design Rationale						
- Project Risks						
- Test Plan, Test Cases						
- Test/Experimental Results						
- User Manual						
- Development Timeline					X	

LEGEND	
Everyone	
Astha	
Daniel	
Mason	

TASKS	Fall Quarter (11/2 - 1/3)						
	Week 7	Week 8	Week 9	Thanksgiving Break	Week 10	Finals Week	Christmas Break
	11/2 - 11/8	11/9 - 11/15	11/16 - 11/22	11/23 - 11/29	11/30 - 12/6	12/7 - 12/13	12/14 - 1/3
Problem Statement							
- Scenarios							
- Technical Questions							
Research							
- SMS application							
- Android application							
- Phone Gap							
- Database/Server							
- Location tracking methods							
Design Document					X		
- Title Page/Table of Cont.					X		
- Requirements					X		
- Conceptual Model					X		
- Use Cases					X		
- Architectural Diagram					X		
- Technologies Used					X		
- Design Rationale					X		
- Project Risks					X		
- Test Plan, Test Cases					X		
- Test/Experimental Results					X		
- User Manual					X		
- Development Timeline							

LEGEND	
Everyone	
Astha	
Daniel	
Mason	

TASKS		Winter Quarter (1/4 - 2/7)				
	Christmas Break	Week 1	Week 2	Week 3	Week 4	Week 5
	12/14 - 1/3	1/4 - 1/10	1/11 - 1/17	1/18 - 1/24	1/25 - 1/31	2/1 - 2/7
Design Review (presentation)				X		
Design Document (revised)					X	
Operational System						
- Frontend (Interface)						
- Health Worker Account						
- Map Display						
- Administrator Account						
- Backend (Server/Database)						
- Testing						

LEGEND	
Everyone	
Astha	
Daniel	
Mason	

TASKS	Winter Quarter (2/8 - 3/28)						
	Week 6	Week 7	Week 8	Week 9	Week 10	Finals Week	Spring Break
	2/8 - 2/14	2/15 - 2/21	2/22 - 2/28	3/1 - 3/7	3/8 - 3/14	3/15 - 3/21	3/22 - 3/28
Design Review (presentation)							
Design Document (revised)							
Operational System					X		
- Frontend (Interface)				X			
- Health Worker Account			X				
- Map Display			X				
- Administrator Account				X			
- Backend (Server/Database)					X		
- Testing					X		

LEGEND	
Everyone	
Astha	
Daniel	
Mason	

TASKS		Spring Quarter (3/29 - 5/2)				
	Spring Break	Week 1	Week 2	Week 3	Week 4	Week 5
	3/22 - 3/28	3/29 - 4/4	4/5 - 4/11	4/12 - 4/18	4/19 - 4/25	4/26 - 5/2
Final Presentation						X
Final Report						
Final Implementation						
- Frontend (Interface)						X
- Health Worker Account			X			
- Map Display				X		
- Administrator Account					X	
- Backend (Server/Database)						
- Testing						

LEGEND	
Everyone	
Astha	
Daniel	
Mason	

TASKS	Spring Quarter (5/3 - 6/11)						
	Week 6	Week 7	Week 8	Week 9	Week 10	Finals Week	Graduation
	5/3 - 5/9	5/10 - 5/16	5/17 - 5/23	5/24 - 5/30	5/31 - 6/6	6/7 - 6/11	6/13
Final Presentation							
Final Report					X		
Final Implementation					X		
- Frontend (Interface)							
- Health Worker Account							
- Map Display							
- Administrator Account							
- Backend (Server/Database)			X				
- Testing					X		

LEGEND	
Everyone	
Astha	
Daniel	
Mason	