

6-9-2016

# SaguaroTrack

Christine Rohacz  
*Santa Clara University*

Kirk Iserman  
*Santa Clara University*

Follow this and additional works at: [https://scholarcommons.scu.edu/cseng\\_senior](https://scholarcommons.scu.edu/cseng_senior)



Part of the [Computer Engineering Commons](#)

---

## Recommended Citation

Rohacz, Christine and Iserman, Kirk, "SaguaroTrack" (2016). *Computer Engineering Senior Theses*. 67.  
[https://scholarcommons.scu.edu/cseng\\_senior/67](https://scholarcommons.scu.edu/cseng_senior/67)

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact [rscroggin@scu.edu](mailto:rscroggin@scu.edu).

**Santa Clara University**  
Department of Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY  
SUPERVISION BY  
Christine Rohacz and Kirk Iserman

ENTITLED

**SaguaroTrack**

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
**BACHELOR OF SCIENCE**  
IN  
**COMPUTER SCIENCE AND ENGINEERING**

*hlu fgu*

---

Thesis Advisor

June 9, 2016

*N. Ling*

---

Department Chair

June 9, 2016

# SAGUAROTRACK

By  
Christine Rohacz and Kirk Iserman

## **SENIOR DESIGN PROJECT REPORT**

Submitted to  
the Department of Computer Science and Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements  
for the degree of  
Bachelor of Science in Computer Science and Engineering

Santa Clara, California

2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	1
1.2	Existing Solutions . . . . .	1
1.3	Our Solution . . . . .	1
1.3.1	BroadbandHoldings, LLC. . . . .	2
<b>2</b>	<b>Requirements</b>	<b>2</b>
2.1	Functional Requirements . . . . .	2
2.2	Nonfunctional Requirements . . . . .	3
2.3	Design Constraints . . . . .	3
<b>3</b>	<b>Use Cases</b>	<b>3</b>
3.1	Administrator Use Cases . . . . .	3
3.2	Field Worker Use Cases . . . . .	5
<b>4</b>	<b>Activity Diagram</b>	<b>7</b>
<b>5</b>	<b>Architectural Diagram</b>	<b>9</b>
<b>6</b>	<b>Web and Mobile Interface</b>	<b>9</b>
6.1	Website User Interface . . . . .	9
6.2	Mobile Application User Interface . . . . .	13
<b>7</b>	<b>Technologies Used</b>	<b>15</b>
<b>8</b>	<b>Design Rationale</b>	<b>17</b>
8.1	Technologies . . . . .	17
8.2	Other Design Considerations . . . . .	18
<b>9</b>	<b>Test Procedures and Test Results</b>	<b>19</b>
9.1	Unit Testing . . . . .	19
9.2	Integration Testing . . . . .	19
9.3	System Testing . . . . .	19
9.4	Deployment Testing . . . . .	20
9.5	Test Results . . . . .	20
9.6	Black Box Testing for User Interface . . . . .	21
<b>10</b>	<b>Risk Analysis</b>	<b>21</b>
<b>11</b>	<b>Development Timeline</b>	<b>21</b>
<b>12</b>	<b>Maintenance Guide</b>	<b>23</b>
12.1	iOS and Android . . . . .	23
12.2	Google Maps API . . . . .	23
12.3	Plugin for Background Operation . . . . .	23

<b>13 Societal Issues</b>	<b>23</b>
13.1 Ethical . . . . .	23
13.2 Social . . . . .	24
13.3 Political/Legal . . . . .	24
13.4 Economic . . . . .	24
13.5 Health and Safety . . . . .	25
13.6 Manufacturability . . . . .	25
13.7 Sustainability . . . . .	25
13.8 Environmental Impact . . . . .	25
13.9 Usability . . . . .	25
13.10 Lifelong Learning . . . . .	25
13.11 Compassion . . . . .	26
<b>14 Conclusions</b>	<b>26</b>
14.1 Summary of Work . . . . .	26
14.2 Lessons Learned . . . . .	26
14.2.1 Developer Biases . . . . .	26
14.2.2 Benefits of Project Planning . . . . .	26
14.2.3 Limitations of Phonegap . . . . .	27
14.3 Suggested Changes and Future Work . . . . .	27
14.3.1 Organize Aggregated Data . . . . .	27
14.3.2 Integrated Devices and Manage Devices Page . . . . .	27
14.3.3 Feature Phone Testing . . . . .	28
<b>15 References</b>	<b>29</b>
<b>16 Appendices</b>	<b>30</b>
16.1 Installation Guide . . . . .	30
16.1.1 Environment Requirements . . . . .	30
16.1.2 Database . . . . .	30
16.1.3 Website . . . . .	32
16.1.4 Mobile Application . . . . .	32
16.1.5 iOS Special Configurations . . . . .	32
16.2 User Manual . . . . .	33
16.2.1 Administrator Web Interface . . . . .	33
16.2.2 Mobile Application . . . . .	34

## List of Figures

1	System Administrator Use Case Diagram . . . . .	5
2	Use Case Diagram for Remote Employee Users . . . . .	7
3	Field Worker Activity Diagram . . . . .	8
4	Architectural Diagram for SaguaroTrack . . . . .	9
5	Login Page . . . . .	10
6	Activity Feed . . . . .	10
7	Activity Feed showing details expanded from a check in status update. .	11
8	List of Devices Page . . . . .	11
9	Map Page . . . . .	12
10	Manage Devices . . . . .	12
11	Login on Mobile Device . . . . .	13
12	Home page for our mobile users from which they can begin tracking or check in. . . . .	14
13	When the user begins tracking, this screen will be displayed until the user chooses to end tracking. . . . .	14
14	When a user is being tracked in the background, updates are continuously sent to the user so they do not forget to end tracking at the end of their work day. . . . .	15
15	Check in form for employees out in the field. . . . .	15
16	Development Timeline . . . . .	22

## List of Tables

1	Risk Analysis . . . . .	21
---	-------------------------	----

## **Abstract**

In an increasingly connected world, many employees work remotely and go out into the field to solve an issue. It is important for their employers to be able to keep track of where their employees are to ensure accountability and quick response times in emergency situations. SaguaroTrack is a new solution to this issue that combines a mobile device application with a Web interface to allow employers to easily account for their mobile workforce. SaguaroTrack will display the location data sent by the mobile device on an easy to understand map to increase the efficiency of response time to customer and company requests.

# **1 Introduction**

## **1.1 Problem Statement**

For companies with a large mobile workforce, holding employees accountable and delivering quality customer service presents many challenges. In many developing countries, such as India, health care organizations rely on sending health workers into the field to assess patients. Unfortunately, many of these companies see low productivity and returns because of high rates of absenteeism. Additionally, due to poorly built infrastructure, members of a mobile task force in rural areas are difficult to contact in the event of a health care emergency. This is a problem that may seem unfamiliar to those living in a developed country, but a similar problem exists for many companies with outreach models. Telecommunications and IT companies send employees into the field every day to serve their clients. Often, a client will call unexpectedly with an urgent request. In such instances, it is important that companies can identify which employees are closest to the client in order to quickly address the client's needs. The common problem of companies needing to know where members of their mobile workforce are at all times arises in several different scenarios.

## **1.2 Existing Solutions**

There currently exist a number of solutions designed to track employee activity in order to dynamically create employee time sheets and record employee gas mileage. Companies like AccuTracking and Xora utilize a tracking system that tags an employee's car or cell phone and monitors his or her GPS location throughout the workday. Unfortunately, these interfaces are poorly designed and difficult for the average user to understand. Other systems such as TSheets and LaborSync are beautifully designed, but focus on creating accurate employee time sheets by allowing employees to easily clock in and out. None of these solutions enable employers to quickly see who is where at any given time during the work day. Additionally, all of these systems are only available at a substantial cost, which is unfeasible for companies in developing nations and undesirable for U.S based companies.

## **1.3 Our Solution**

We propose a user-friendly, web-based system that can easily display the location of all mobile workforce members with a mobile application to collect that data. In the event of unexpected occurrences such as health care emergencies or sudden customer demands, companies will be able to readily locate the field employees closest to that event. The mobile application will track the GPS coordinates of a mobile device as the employees visit field locations. When network connectivity is available, the data collected on the device will be uploaded in real time to a central server and displayed on a web interface available for office employees and managers to view. There will also be an available option for the field employees to check in at certain locations, like a customer's home, for managers to easily monitor their attendance. While the system displays a general map of the entire mobile workforce, a system administrator will be able to query an area in order to quickly determine which employees are closest. The system will be a general platform available on most mobile platforms that is highly configurable depending on



unique business needs. We seek to curb high rates of absenteeism in some countries and help address customer service demands in others.

### **1.3.1 BroadbandHoldings, LLC.**

BroadbandHoldings is a telecommunications company based in Scottsdale, Arizona that specializes in DirecTV commercial property installs. The company has a large amount of employees who go into the field every day to install and repair DirecTV dishes. They have expressed an interest in having a solution that allows them to keep track of where members of their mobile workforce are and how many man hours their employees spend on a job. We have worked with BroadbandHoldings to develop a solution that will help them address these problems and have tested our solution with their company in the field. As a result, working with the company has helped us develop generic requirements that may suit a company like BroadbandHoldings.

## **2 Requirements**

We gathered the requirements for our system by talking to potential users of the system. We gained feedback from potential users at BroadbandHoldings about what requirements the system must have and what features they would like to see. Feedback was formalized into functional and nonfunctional requirements as well as design constraints as described below.

According to Essentials of Software Engineering, functional requirements describe what a program needs to do and nonfunctional requirements describe the manner in which the functional requirements need to be achieved. Functional requirements can normally be measured by either true or false: either the system does or does not implement a functional requirement. A nonfunctional requirement however though usually relates to issues of performance, reliability, and security, and are usually measured by degree of satisfaction. Along with the system's requirements, we identify design constraints. These are normally constraints imposed by availability of technology or requirements the client/customer has [9].

### **2.1 Functional Requirements**

- The system will automatically gather GPS data from mobile devices in specified time intervals
- The system will display the location of all the users who are currently being tracked by a mobile device through a Web interface
- The system will allow users to “check in” their location (manually send GPS data)
- The system will allow users to add client information and feedback when checking in
- The system will allow users to query an area for the closest mobile devices
- The system will allow administrators to search for a specific employee who is currently being tracked
- The system will allow administrators to view employees' recent trips and locations

## 2.2 Nonfunctional Requirements

- The system will be user friendly in order to accommodate individuals from various technological backgrounds
- The system will be built to scale for a flexible number of mobile users
- The system should efficiently consume the battery of the mobile device
- The system will provide sufficient documentation to help users understand the application

## 2.3 Design Constraints

- The system should have cross browser compatibility for major Internet browsers
- The system should be available on various mobile platforms
- The website should run on a secure sockets layer (SSL) connection (required for Chrome web browsers)

# 3 Use Cases

Use case diagrams provide a simple diagram that models the different types of users who interact with our system (known as actors) and the goals each user wishes to achieve. Our system has two intended users: the administrators who monitor remote employees, and the employees themselves who use the system to check in at various locations and record customer feedback.

## 3.1 Administrator Use Cases

Figure 1 shows the use case diagram for a system administrator in SaguaroTrack. Further details for the administrator's use cases are described below:

### Login

- Goal: Gain administrator access to SaguaroTrack
- Precondition: Have valid username and password
- Post-condition: Admin is logged into the system
- Steps
  1. Access the webpage
  2. Enter username and password
  3. Submit credentials
  4. Redirect to home page
- Exceptions
  - Invalid username or password

## **View List of Field Workers**

- Goal: View list of all field workers with assigned devices
- Preconditions: User must be logged in
- Post-conditions: List of all employees with tracking ID's is displayed
- Steps:
  1. Select option to view devices

## **Assign Devices**

- Goal: Assign a new tracking ID to an employee
- Preconditions: Employee does not already have a tracking ID
- Post-condition: Employee has assigned unique tracking ID
- Steps:
  1. Select option to register new device
  2. Fill out all relevant fields for new device
  3. Submit information
- Exceptions
  - User has already been assigned tracking ID or user already exists

## **View Real Time Map of Device Positions**

- Goal: View map of real time device locations
- Preconditions: User must be logged in
- Post-condition: Map displays location of all currently live devices
- Steps
  1. Select option to view map
  2. Map is populated with list of devices currently being tracked
- Exceptions
  - Browser cannot reach servers that hosts Google Maps API

## View Device Track History

- Goal: User wants to view the specific tracking history for a single employee
- Preconditions: User is logged in and employee has assigned tracking ID
- Post-conditions: Tracking data related to employee is displayed
- Steps:
  1. From the list of assigned devices, select a specific device/user from which to view information about
  2. View the recent activity in the user's profile
  3. Select an activity from the "recent activity" table to get more details

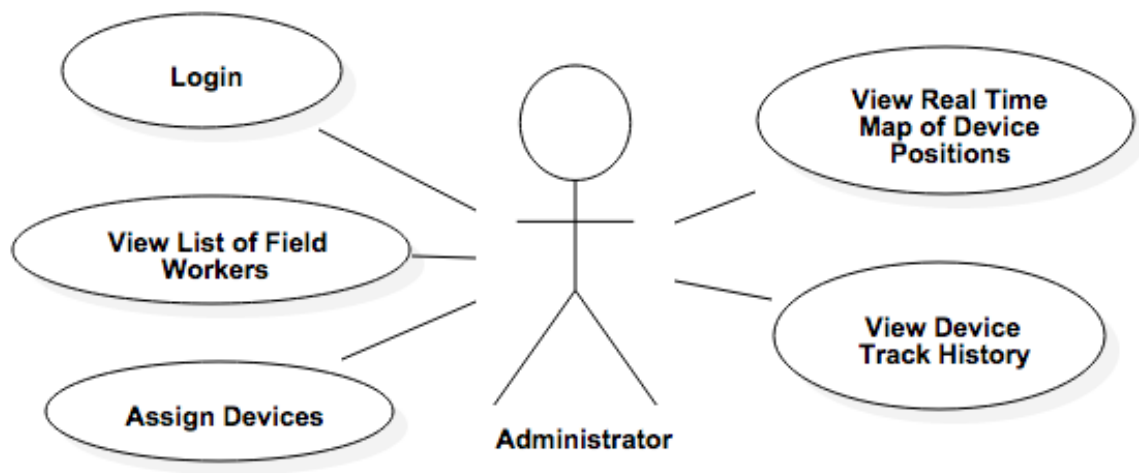


Figure 1: System Administrator Use Case Diagram

## 3.2 Field Worker Use Cases

Figure 2 shows all the goals a field worker wishes to achieve using the system. More details for each goal are described below.

### Login

- Goal: Associate device with assigned tracking ID
- Precondition: Device has no currently associated tracking ID
- Post-condition: Device now has associated tracking ID
- Steps
  1. Open mobile application

2. Enter tracking ID
  3. Submit
- Exceptions
    - Invalid tracking ID entered

### **Begin Tracking**

- Goal: Start automatic tracking which monitors the users movement automatically
- Preconditions: User has application installed on their mobile device
- Post-condition: Automatic tracking has begun
- Steps:
  1. Select option to begin tracking

### **End Tracking**

- Goal: End tracking of device
- Preconditions: Tracking is currently in progress
- Post-condition: Tracking has ended
- Steps
  1. Select option to end tracking

### **Check In**

- Goal: User wants to check in at specific point to record their location
- Preconditions: Tracking has started
- Post-conditions: User's location data is gathered and recorded
- Steps:
  1. Select option to check in

### **Record Feedback**

- Goal: When checking in, user wants to record feedback with check in location
- Preconditions: User has checked in
- Post-conditions: Information is recorded with the user's check in location
- Steps:
  1. Fill out input fields regarding the user's current location

2. Submit information

- Exceptions
  - There is no feedback to record

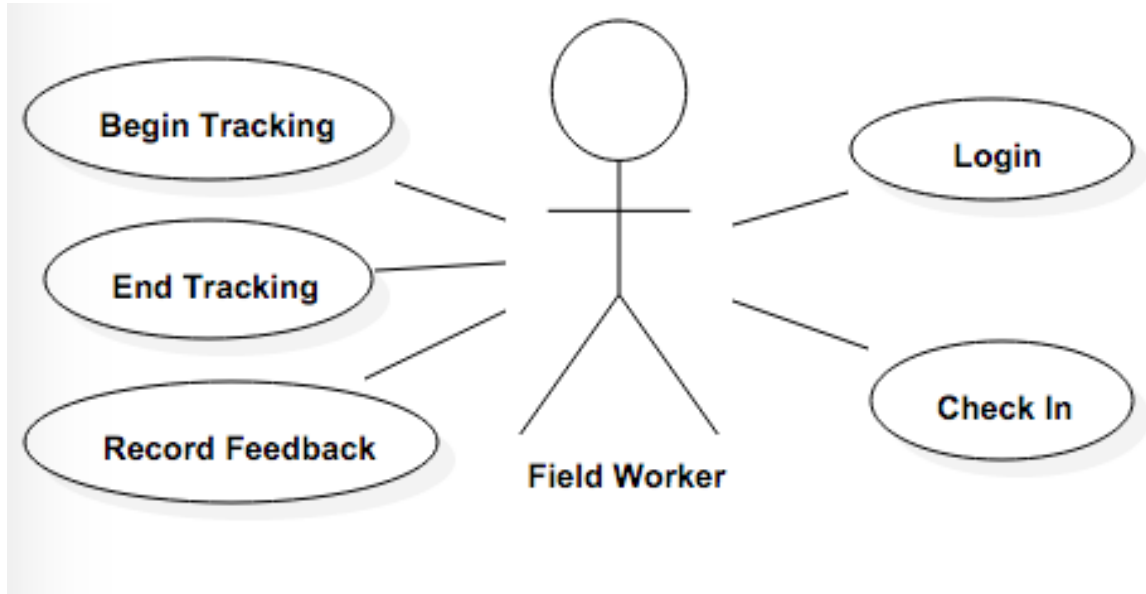


Figure 2: Use Case Diagram for Remote Employee Users

## 4 Activity Diagram

Activity diagrams help to visualize how the system works by showing the general flow of how tasks would be accomplished by different users. This document provides an activity diagram only for the field workers because the activity diagrams for administrators is too complex. More details about the administrator's interactions with the system can be seen in website and user interface designs in Section 6.

Figure 3 displays the activity diagram for an employee who uses the mobile application to track their positioning. First, the employee enters in their ID to login and then selects "Begin Tracking." From here, the employee has the option to end the tracking, or to check in and record feedback from a customer. After ending tracking or checking in, the user has the option to log out.

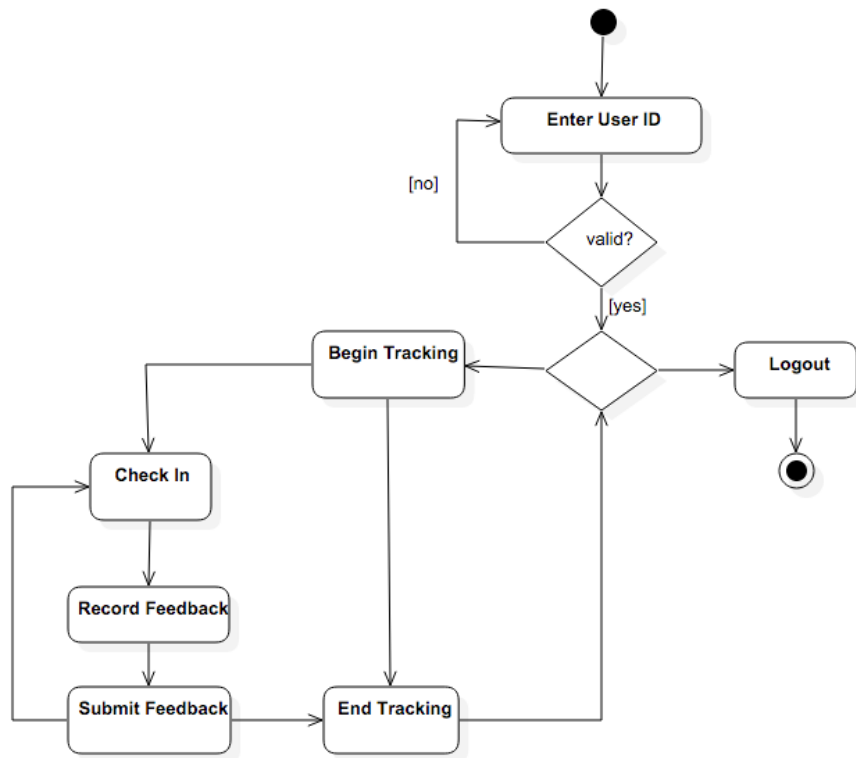


Figure 3: Field Worker Activity Diagram

## 5 Architectural Diagram

An architectural diagram shows an overview of the major components of the system. It does not go into much detail, but rather shows just the largest components of the system and how they interact with each other.

Our system uses a variation of the client-server model. There are two interfaces: the administrator web interface and the employee tracking device interface. Connecting the two is the server and data storage, which holds the information relayed by the employee device, and also runs the programming logic to display this information for the administrators on the web side. The mobile application communicates with the server and database through AJAX requests. The web interface however, uses PHP and MySQL to communicate with the database.

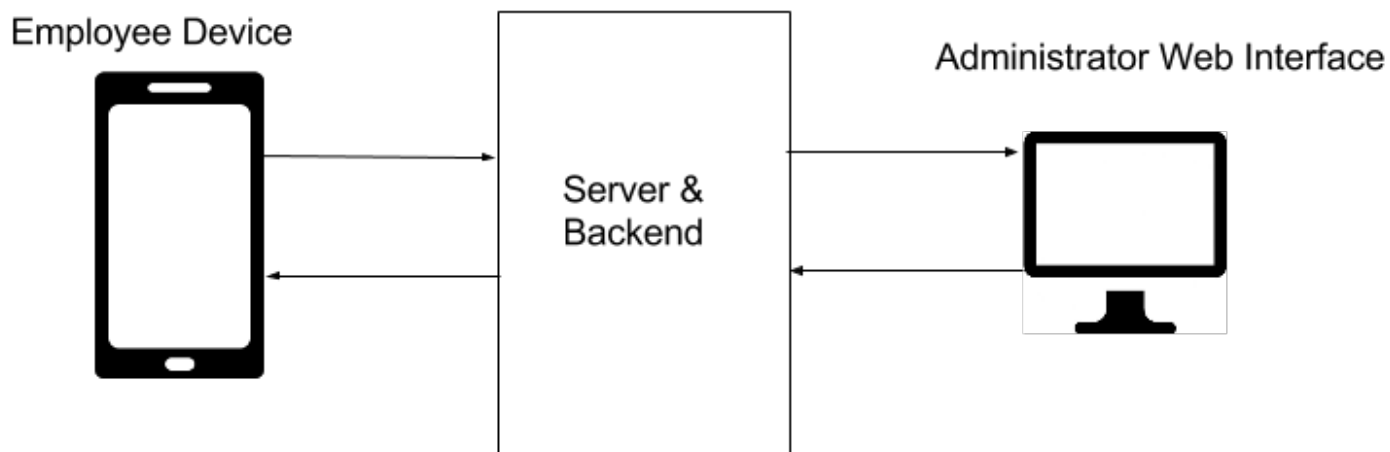


Figure 4: Architectural Diagram for SaguaroTrack

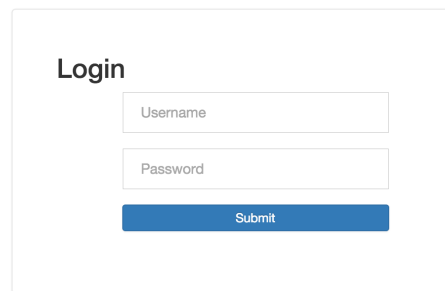
## 6 Web and Mobile Interface

Images of the web and mobile interface are shown in the following sections to describe our system's conceptual model.

### 6.1 Website User Interface

Figure 5 shows the administrator login portal, this is where user's gain access to our website. This is the first page that will be displayed to the user when they access the given web URL for the website and they must have valid credentials for logging in.



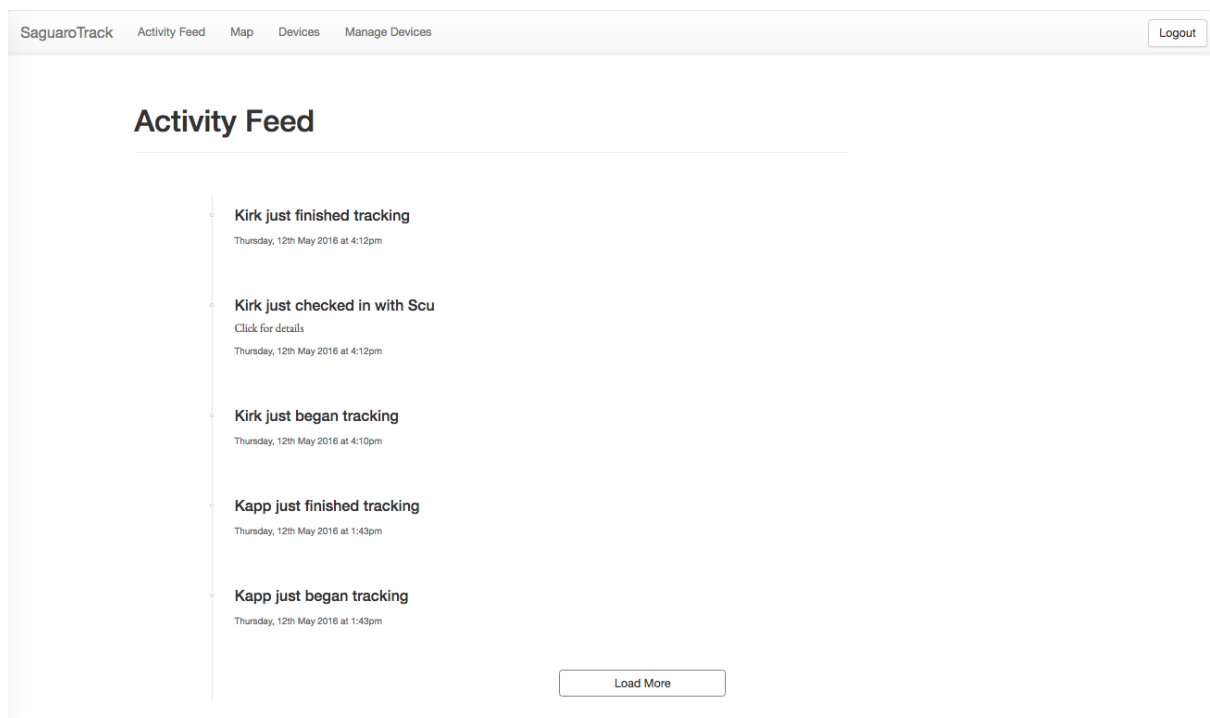


The login form is centered on the page. It has a title 'Login' at the top. Below the title are two input fields: 'Username' and 'Password'. At the bottom of the form is a blue 'Submit' button.

Figure 5: Login Page

Figure 6 shows a detailed activity feed for recent events received from various employee devices. This is to give the user a macroscopic overview of all activity from employees in the field. This page includes status updates such as whenever a user begins tracking, ends tracking or checks in a specific location.

For more details about a specific event, "Click for details" can be clicked to expand more information about an update as shown in Figure 7



The Activity Feed page has a navigation bar at the top with links: SaguaroTrack, Activity Feed, Map, Devices, and Manage Devices. A Logout button is in the top right corner. The main heading is 'Activity Feed'. Below it is a vertical timeline of events:

- Kirk just finished tracking**  
Thursday, 12th May 2016 at 4:12pm
- Kirk just checked in with Scu**  
Click for details  
Thursday, 12th May 2016 at 4:12pm
- Kirk just began tracking**  
Thursday, 12th May 2016 at 4:10pm
- Kapp just finished tracking**  
Thursday, 12th May 2016 at 1:43pm
- Kapp just began tracking**  
Thursday, 12th May 2016 at 1:43pm

At the bottom of the feed is a 'Load More' button.

Figure 6: Activity Feed

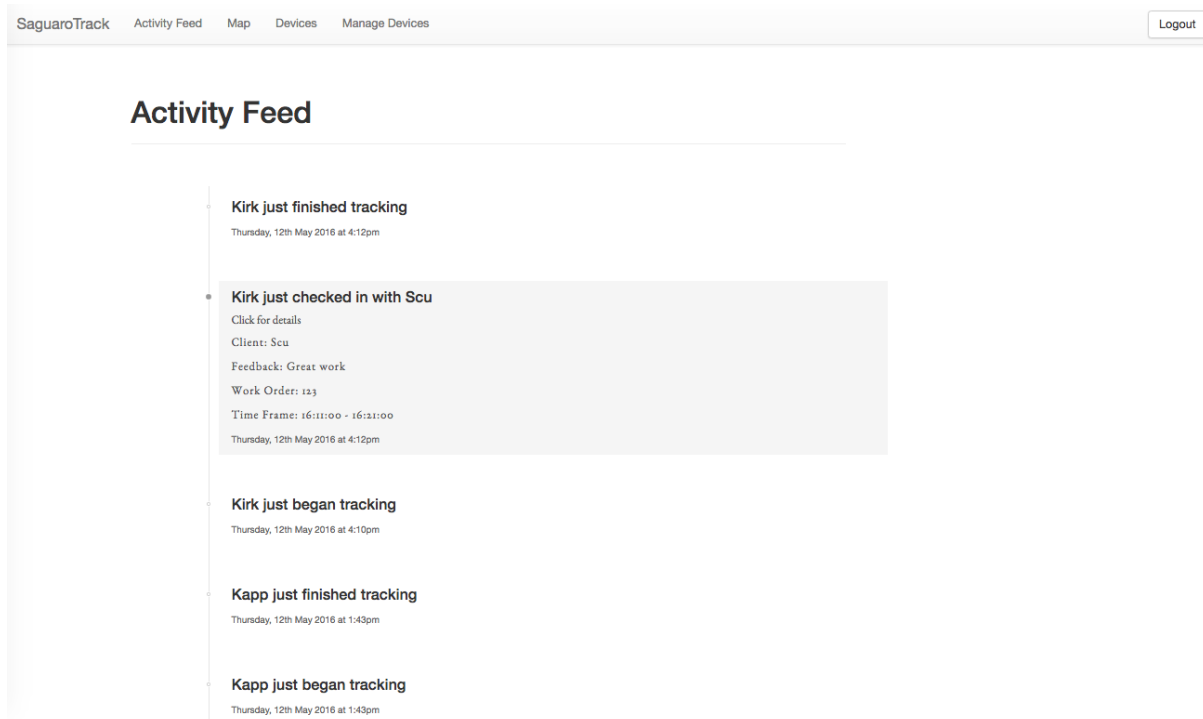


Figure 7: Activity Feed showing details expanded from a check in status update.

Figure 8 shows a list of employee devices in the left hand column. Clicking on one of those devices brings up the more detailed frame in the center that contains information about that particular device, such as who it belongs to, and recent activity from that device. A user will be allowed to select a date from an input field and the Recent Activity list on the right side of the screen will be filtered to show just events from that date.

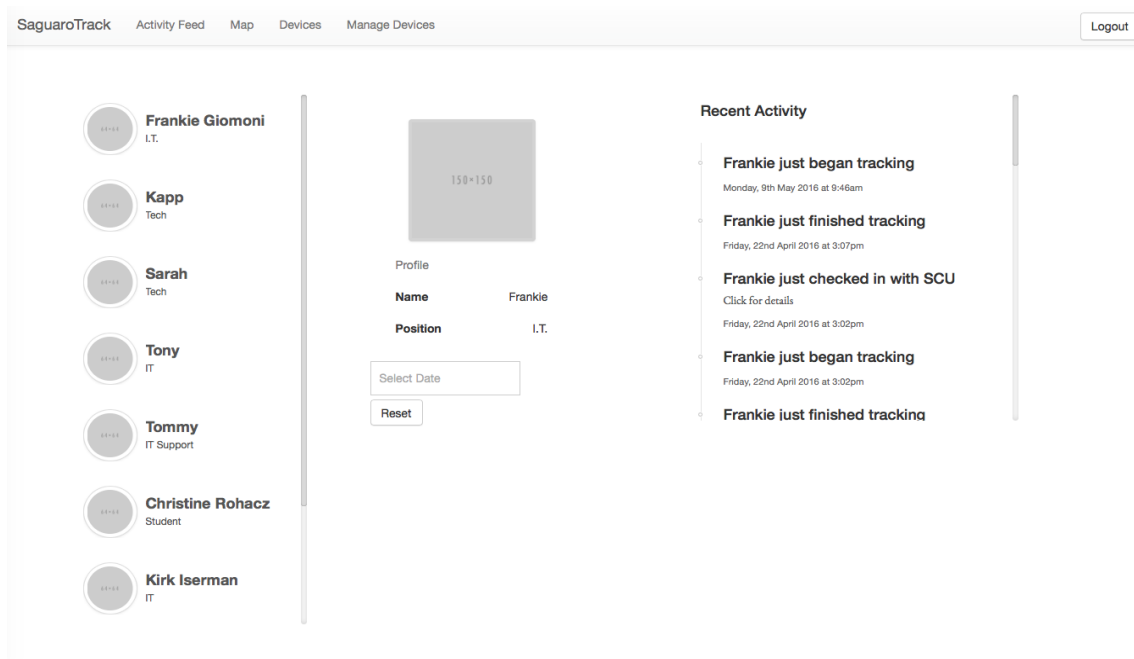


Figure 8: List of Devices Page

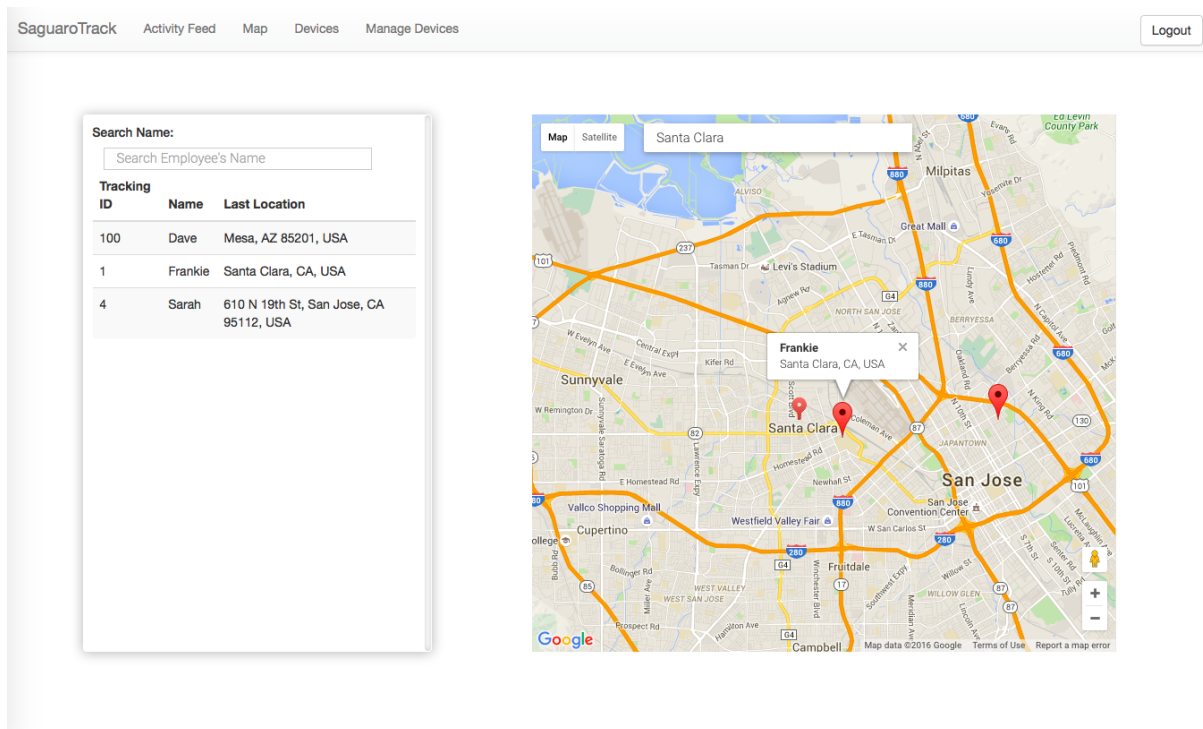


Figure 9: Map Page

Figure 9 shows a map with markers that identify the locations of any currently active devices. When we click on a marker, we can see the name associate with that position. Lastly, Figure 10 shows the Manage Devices page which allows an administrator to manage the mobile devices, such as adding and removing a device or changing information associated with a device. This page is intended to help companies that assign company phones to their employees and may need to update them often within the system.

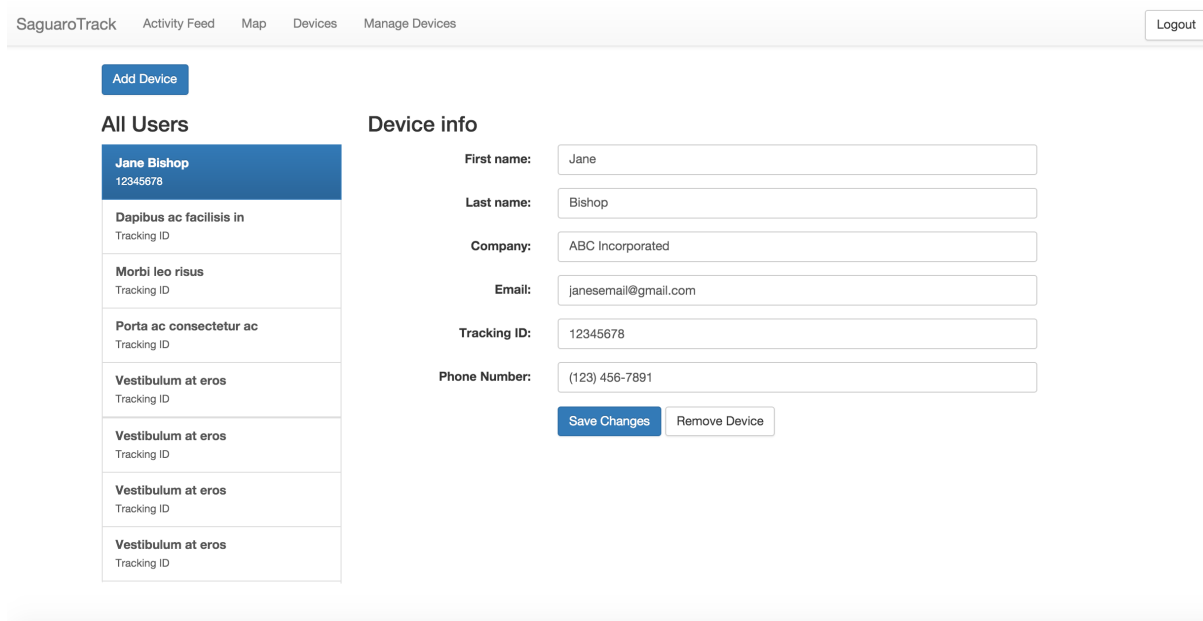


Figure 10: Manage Devices

## 6.2 Mobile Application User Interface

Employees who are being tracked will have a mobile application running on their device. When they first open the application, the screen will prompt them to “Login” with their tracking ID as seen in Figure 11. This is an ID that is assigned to them through a system administrator and is then associated with their user profile and device.

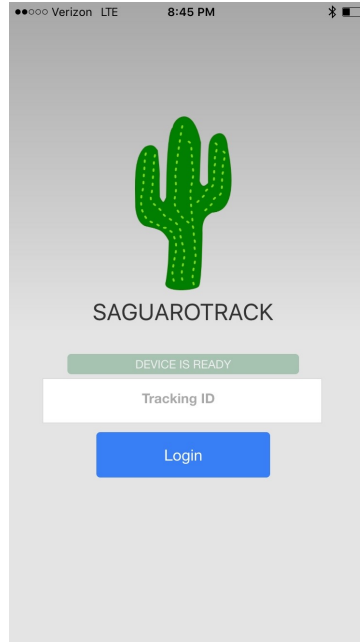


Figure 11: Login on Mobile Device

After a user has logged in, they will be prompted with the option to begin tracking as shown in Figure 12. Once tracking begins, the user will be tracked and their device will be pinged at random intervals to gather their location data. Simultaneously, the user will be given the option to check in whenever they wish. The user's location will always be displayed to the user and the user can choose when to end tracking as shown in Figure 13.

When a user is being tracked, location updates will be shown in the background so that the user does not forget he or she is being tracked. This also serves as a reminder for users to remember to end tracking at the end of their work day. An example of our background reminders can be shown in Figure 14.

When a user chooses to check in, the user will be shown the feedback form in Figure 15. The user will submit this form once it is filled out and the screen will return to the previous tracking screen, shown in Figure 13. Whenever a user has finished their work schedule for the day, they will be able to end tracking and logout.

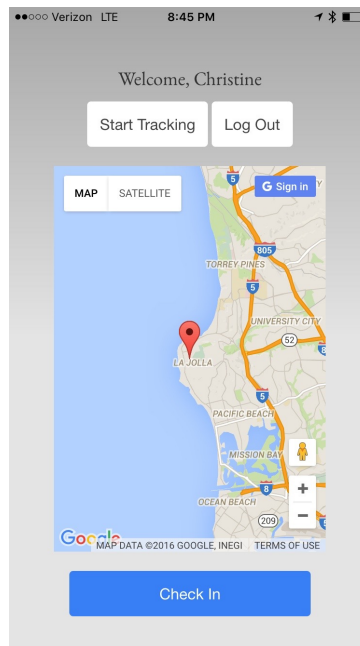


Figure 12: Home page for our mobile users from which they can begin tracking or check in.

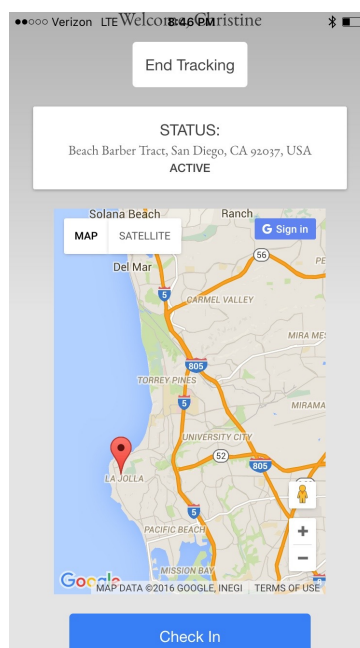


Figure 13: When the user begins tracking, this screen will be displayed until the user chooses to end tracking.

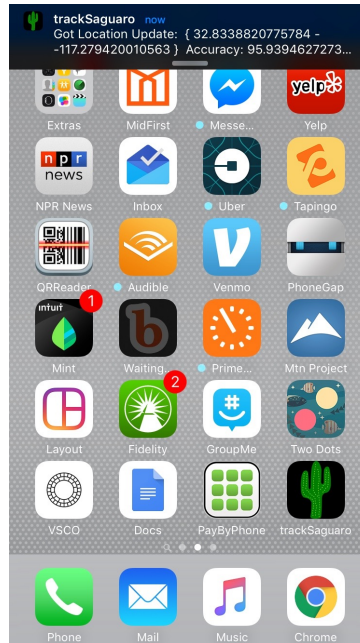


Figure 14: When a user is being tracked in the background, updates are continuously sent to the user so they do not forget to end tracking at the end of their work day.

Figure 15: Check in form for employees out in the field.

## 7 Technologies Used

**HTML** Known as hypertext markup language, HTML is used to create web pages and is a useful foundation for all websites. Through HTML, web pages can embed or refer to CSS and Javascript code

**Javascript** Javascript is supported by all major web browsers and is a scripting language commonly used to create interactive content within a website.

**AJAX** AJAX stands for asynchronous Javascript and XML, and is a set of web development techniques used on the client-side to create asynchronous Web applications.

**CSS** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language, such as HTML. It is most commonly used to visually stylize a Web page.

**PHP** PHP is a common server-side scripting language designed specifically for Web development.

**Github** GitHub is web-based graphical user interface to Git, which is a version control system. Git is used to keep track of numerous revisions made to coding projects, maintain documentation, keep track of bugs, and allow programmers to easily collaborate without interfering on each other's work.

**LaTeX** LaTeX is a typesetting language that extends the typesetting system TeX. It is primarily used for designing mathematics, science, and engineering documents.

**MySQL** MySQL is an open source relational database management system. It is a popular choice for database use in many web applications and allows data to be stored and managed through a relational database management system.

**Google Maps Javascript API** Allows developers to add a unique, customized map to their webpage. As of May 2016, Google Maps API requires that geolocation requests sent to the API come from a SSL (Secure Sockets Layer) URLs (commonly known as https URLs). This requirement is to ensure that locations are being shared across a secure connection.

**Google Maps Geocoding API** Geocoding is the process of converting physical addresses into geographic coordinates (latitude and longitude values). This API also supports reverse geocoding which converts addresses vice versa. The API is intended to geocode already known addresses for placement of content on a map.

**cordova-background-geolocation-services** The Cordova background geolocation plugin was created to provide normalized background geolocation updates for both Android and iOS which both require device-specific implementations to implement background geolocation. This plugin is in active development and is continuously updated with android and iOS platform updates. It is also backwards compatible with previous versions of each platform.

## 8 Design Rationale

### 8.1 Technologies

**HTML/CSS** HTML and CSS are the standard in modern Web programming, and are easy to use to create user-friendly websites.

**Javascript** Javascript is used to interface with the Google Maps APIs to display the location of employees. It is also used for some initial client-side validation as well as general interactivity for the website.

**AJAX** AJAX is used to dynamically update portions of web pages. For example, when a mobile device sends new data to the servers, the Map page as seen in Figure 9 will automatically update with the new location displayed on the map as well as the recent activity section. Additionally, AJAX is used by our phonegap mobile application in order to send data between the application and our MySQL database. Phonegap does not inherently support PHP which is normally used for communicating with MySQL, so AJAX is used instead.

**PHP** PHP will be used for any database interfacing and data processing, and it lends itself well to representing the data with a webpage. Data is normally stored in tables and it is easy for our system to large amounts of data with little delay. Additionally, PHP is well known by the developer community and should our application be continued on or used by another team, it will be easy for other developers to pick up on our methods.

**Github** We will use Github for version control, as a code backup system, and for organizing the file hierarchy. When changes are made, we commit these changes to our Github repository. In the future, if one change were to break the entire system, we can pull a historic version of the repository, one that we know was working, and begin working from that one. Github also prevents the possibility that we will lose our code by damaging our computers because it allows us to backup our code consistently and frequently.

**LaTeX** LaTeX provides a simple way to cleanly portray our documentation without having to worry about spacing or formatting issues. It allows to focus on the quality of the content rather than the presentation of it. Without Latex, this document in your hands would not look nearly as flattering.

**Phonegap** A easy to use framework for developing hybrid mobile applications in HTML5, CSS and Javascript. A hybrid mobile application is one that works on every device, including feature phones and blackberry. Phonegap allows our app to maintain backwards compatibility (i.e. working on older versions of a device) and also allows us to build the app for a wide community of users.

**Ionic** Ionic is a front-end framework for developing hybrid mobile apps with HTML5. It is well suited to working with phonegap but we primarily used Ionic to design a smooth user interface for the mobile application.



**MySQL** The MySQL database system is easy to understand and use. PHP also interfaces well with MySQL to make the data management easier. Additionally, the data we collect from our mobile devices is well suited to a relational database format which displays data in the form of tables. What this means is that large amounts of data can be grabbed from our database in a relatively short period of time.

**Google Maps Javascript API** The Javascript API is used on the web side to show a visual map of a location, along with markers for the employee locations.

**Google Maps Geolocation API** The Geolocation API is used on the mobile device application to obtain the location of the device based on information gathered from nearby cell towers and WiFi access points.

**Google Maps Geocoding API** The Geocoding API will be used for translating the raw latitude and longitude coordinates into physical addresses to display content in a more user-friendly manner.

**Background Geolocation Plugin** The reason we chose to use a third party plugin to enable background geolocation services was to enable our services on both iOS and Android devices in the little amount of time we had. When we originally developed our app, the background geolocation was working on Android but not on iOS and in order to address this issue in just the few weeks we had left, we choose to leverage the use of a third party plugin to speed up development which would allow us deploy the application on both android and iOS devices. The plugin and its author is referenced in our code.

## 8.2 Other Design Considerations

**Platform** We initially chose to develop the mobile application for the Android platform for several reasons. Android devices are much more common than Apple devices or other smart phone platforms, so it is a larger target market overall. Additionally, it is easier to test and develop for Android devices than other platforms because no developer license that needs to be purchased, and the Android operating system allows for application installation from locations outside the built in application store.

However, we learned that the company we were working with, Broadbandholdings, had iOS company devices that our application would be installed on. We decided to expand and take full advantage of the multiplatform capabilities offered by Phonegap by creating the iOS application as well for our deployment testing.

**Web Interface** With a Web interface, the employers/employees do not need to download any additional software other than the mobile application to operate the system. The employers or administrators in charge of overseeing the field employees are free to use a variety of browsers on different types of devices to accomplish their tasks. The user interface will be the same across all of these devices. From a development perspective, this makes the system easier to develop and maintain because there is only one platform to manage.

**Devices** Our web interface is supposed to give employers enough information about devices that they could easily determine where employees were and when. In order to do this, we created a "Devices" page which lists all the devices that currently have tracking IDs associated with them. This page displays device-specific data such as location or check in data that is only associated with a certain device.

**Security** We choose to use a tracking ID with our system to improve security. Instead of sending personal information, like a user's name or phone number, over the network, we send the tracking IDs as a form of employee identification. We are also taking advantage of the fact that Phonegap leverages Web technologies by using SSL to automatically encrypt all data sent between the server and the mobile application. In addition to using SSL, we encrypt our data at rest in our MySQL database by using the built in encrypt/decrypt functionalities of MySQL. Finally, we do not store the actual plaintext of a password to login to the system, but rather stores a cryptograph hash of the password.

## 9 Test Procedures and Test Results

### 9.1 Unit Testing

Throughout development, we continuously ran unit tests to assess the smallest testable portion of our system. Conducting unit testing during development is sometimes referred to as test-driven development (TDD). Although TDD can make the implementation phase more tedious, it is worthwhile because it saves time in the long term. It might not be possible to test a unit for every possible scenario, so we mainly focused our testing on general cases and edge cases. We wrote unit tests every time: a) Changes were made to the system and it needed to be verified that the changes did not break any existing code. b) New functionality was added.

### 9.2 Integration Testing

When unit tests were finished, we began integration testing to assess the interface between individual units. Integration testing is intended to identify the problems that occur when unit tests are combined. To develop integration tests, we identified various different subsystems and elements to be integrated into each subsystem. Each unit was already tested individually, so we knew that any errors that arose during integration testing indicated an error between components and not within them. The following are some modules we tested in integration testing: a) Is this user's location displayed correctly on a map? b) Can the system determine a user's locations at a single point? c) Can a user start tracking? d) Can a user check in at a location? e) Does the system fetch and store data in the database correctly? Our integration tests focused on entering valid and invalid data in the system to see how it responded to different inputs.

### 9.3 System Testing

System testing ensures that the integrated system meets all of the pre-established requirements. We conducted two main types of tests during this phase: black box and white box. Black box testing is done with no knowledge of the code base, whereas white

box testing is performed with knowledge of the code and uses that knowledge as a basis for designing test cases. We performed system testing on both Android and iOS devices. iOS devices require a developer certificate from Apple as well as a physical Apple device, whereas Android device testing just requires an Android device, with no developer licensing.

**White Box Testing** For white box testing, we ran tests to examine the system’s ability to trace a mobile device’s path and the accuracy of that functionality. We also tested various inputs and edge cases to our web interface to ensure that it properly performed all of those requests.

**Black Box Testing** We engaged in black box testing by letting potential users of our system participate in certain scenarios that tested different requirements, otherwise known as acceptance testing. These tests are used to gauge the system’s level of usability and user friendliness. We collected feedback from our users about improvements that should be made to the system or which features might have been unclear.

## 9.4 Deployment Testing

After our system was near complete, we visited Broadbandholdings, LLC to test our system in a business scenario. Broadbandholdings expressed interest in using an app that would allow them to track their technicians who go into the field daily to install telecommunications equipment across the U.S. Given the geographic diversity of their mobile workforce, the company was a perfect candidate for testing our solution. To begin testing our application, we deployed it on their employees’ phones and asked individuals to provide us feedback on what they thought. Our system’s full functionality was tested and we were able to see how it responded in a real-world situation.

## 9.5 Test Results

As expected, the thorough unit testing helped to minimize errors while performing Integration and System Testing. Through our white box system testing, we discovered some technical issues with our system not performing exactly as it should have, but we resolved those errors. We received helpful feedback from our Deployment Testing with Broadband Holdings to direct improvements.

We tested the app with three technicians working on projects in Arizona and did black box testing with employees in the office. We initially received some concerns from the technicians about being tracked throughout their day, but after explaining the privacy policy and security features, we were able to mitigate their concerns. Over time, the employees became more comfortable with the application. The technicians found the app and its interface very easy to use. They reported no problems with it or submitting data. The server-side however, encountered some errors during the testing.

Overall, the functionality of the app worked and connected with the server to send data. However, during our own white box testing, we failed to account for the fact that the app kept falling asleep in the background and could not be woken by the CPU to continue to collect location data. Location data we received was extremely sparse and only occurred when the user opened the app, or shortly thereafter. In addition, Android and iOS require a slightly different work around to allow the application to run

in the background in case the users wished to use other applications; therefore, device independent solutions had to be designed. Our app functioned properly otherwise.

## 9.6 Black Box Testing for User Interface

Broadband Holdings expressed interest in more automated analysis of the data that is gathered and to display that aggregated data in a variety of mediums. Additionally, users reported being unclear as to why there were two separate screens to see information about individual devices. Our system has one webpage to see details about the device and its tracking history and another webpage to change user information associated with the device. Employees at Broadband Holdings recommended that we combined these two pages to make the user experience more fluid. They also recommended we added another layer of security for editing, adding, or deleting device information. Overall, the employees at Broadband Holdings said that the user interface for both the application and the website were very friendly and intuitive.

To address the feedback provided during our testing, we implemented the extra layer of security that was recommended on top of our Manage Devices page and started work on displaying aggregated data more clearly. We are collecting other data that could be useful to companies, but we are not yet displaying on the website due to time limitations. More information about suggested improvements can be found in Section 15.

## 10 Risk Analysis

A risk analysis highlights some of the major potential risks that could hinder the project. It shows the overall impact(calculated by probability\*severity) that the particular risk carries, as well as strategies to help mitigate the impact of each risk.

Table 1: Risk Analysis

Risk	Consequences	Probability	Severity	Impact	Mitigation Strategies
Bugs	Potentially unresponsive program	0.95	4	3.8	Thorough testing, good coding practices
Incorrect Requirements	Delay, Redo sections of the project	0.5	5	3	Design review, know when the set deadlines are
Failure to meet deadlines	Product unfinished	0.05	9	0.45	Follow development timeline, know when set deadlines are
Data loss	Delay, redo work	0.01	7	0.07	Backup in cloud and locally

## 11 Development Timeline

Figure 16 shows a general overview of our development timeline. The components identified in the timeline are worked on by both members of the group and broken down into even smaller components in separate documents for our advantage.

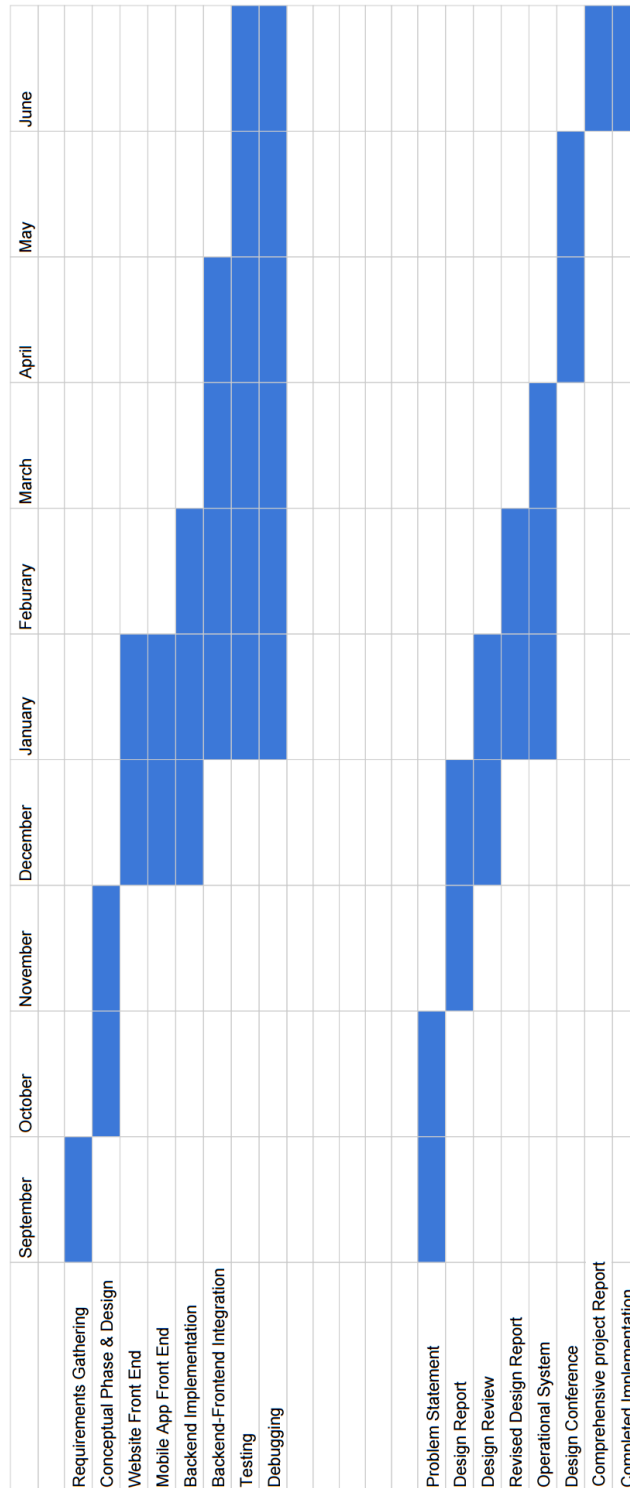


Figure 16: Development Timeline

## **12 Maintenance Guide**

### **12.1 iOS and Android**

As updates are released for both the iOS and Android mobile operating systems, we need to make sure that our application is still compatible. Various security features might be implemented in the future that limit GPS tracking, so our application may need to change to conform to new standards. Also, we need to make sure that our application is also backwards compatible in case users are using older versions of the operating systems.

### **12.2 Google Maps API**

As Google Maps updates and is changed by Google, the API, or Application Program Interface, is likely to change as well. Once again, we need to make sure that all of our calls to the Google Maps API stay up to date and compatible with the latest version so that our application works properly. Even during our development, some of the Google Maps function calls became deprecated, so we had to update our system to the new functions.

### **12.3 Plugin for Background Operation**

As mentioned above in the Technologies section, we ended up using a third party plugin to help wake up our application and allow to run in the background of iOS and Android devices. As new versions of iOS and Android are released, we will have to update this plugin to be compatible. This might be done by the plugin's creator or by us. Eventually, we wish to move away from needing third party support to allow this functionality.

## **13 Societal Issues**

### **13.1 Ethical**

The major ethical concern associated with our project is the fact that we are gathering the location data of individual people and storing that information in a database. Obviously, if this is not executed properly, it could be viewed as a violation of an individual's privacy. According to McNamee, Michael, and Michael, "most ethical issues are connected to the control aspect of GPS tracking, as it imposes an intrusive method of supervision." [6] With this in mind, we designed our application to allow the users who are actually being tracked to maintain as much control over when they are being tracked as possible. Within our mobile application, the employees have control of when they start and stop the tracking, as it is not the goal of our application to track people during their off time. We aim to track them during work hours for the purpose of improving whatever company they work for, not for the sake of always watching their movements. As an example, after Automated Waste Disposal Incorporated started using GPS tracking on its employees, "the number of overtime hours dropped from 300 to 70 hours on average per week." [6] As can be seen from this example, using GPS tracking during work hours benefitted the company by cutting down on unnecessary overtime pay. As mentioned before, we only wish to improve the business practices of companies, not to invasively track their employees, and we do this by allowing the employee who is actually being tracked to have full control over when the application is monitoring them.

## 13.2 Social

As our application is a tracking application, it could have a potentially large social impact. The section above mentions the ethical considerations of tracking, and the section below mentions the legal considerations of tracking. During our deployment testing, several of the employees we were working with were wary of being tracked. However, as more tracking applications like ours are introduced to society, people may become more comfortable with the notion that their location is being recorded. This could be good or bad, depending on how the technology is used.

## 13.3 Political/Legal

The use of mobile tracking to monitor employees' locations during work hours is a relatively new use of technology, and therefore there have not been many court cases or legal decisions made about whether it is constitutional or not. There have, however, been some court cases that clarify when tracking is and is not legal. In *Cunningham v. New York Department of Labor* (NY Ct. App. 2013), the court came to the conclusion that tracking must be done within business hours and tracking data gathered outside of regular business hours is considered unreasonable in the event that it is used to hold an employee accountable. Additionally, the Supreme Court in *City of Ontario, California v. Quon* upheld that information gathered from company owned vehicle or devices is constitutional, and does not invade the privacy rights of employees. Each state has specific laws regarding the tracking of employees. California, for example, has ruled in California Penal Code §637.7 that "No person or entity in this state shall use an electronic tracking device to determine the location or movement of a person," except in the event that an employee gives explicit consent [3]. This is relevant for companies who allow employees to bring their own devices to work, also known as BYOD. Companies that allow this have employees sign BYOD policies. If a company wishes to track personal devices of employees, then it should be explicitly be stated in the BYOD policy so that the employees are aware.

## 13.4 Economic

We had very few economic concerns arise in our project, the only exception being an Apple Developer Account. Apple requires you to pay \$99 a year to support a developer account, which allows you to test your project on an iOS device and to submit your project to the Apple store for approval. Other than that, the code we built ourselves did not require raising any money or taking out a loan. This is one of the largest benefits of the field of Computer Engineering in that many of the tools needed to build software are free and open source. However, actually deploying the app would cost money. The first cost would be the cost of having our application be approved by Apple for publication on the App store. The similar approval process for Android devices does not cost anything. Additionally, we would need to pay for a server to host our project, and depending on how large the project grows, the cost of server space might grow. These costs could be easily offset by charging for the use of our application.

### **13.5 Health and Safety**

No health effects arise from our project, but safety is a primary concern of ours. Related privacy issues are described in the political and legal section above. The location of those who are being tracked is our main safety concern because we need to protect their identity and location. For this reason, our application only passes numbers and does not pass any personal identifying information over the network. This way, if sent data is intercepted or stolen, and the encryption is broken, it is not obvious to whom that data belongs. Personal identifying information is only available to those who have access to our database. To prevent access to our database from the website, we have sanitized all form inputs to ensure that no hackers can steal this information.

### **13.6 Manufacturability**

Our product is not one that has any physical components that need to be manufactured as it is solely code based. The costs of deployment are discussed above in the Economic section

### **13.7 Sustainability**

Our product does not have any direct limited resources that it requires, other than the energy to power the devices it is running on. However, a sustainability issue that may arise with our code is the fact that it does need to be kept up to date with the environments that it is running on. Whenever an iOS or Android update is released, we may need to update our code to ensure that our application is still compatible and functioning.

### **13.8 Environmental Impact**

The only resources that our product needs is the hardware of the servers to store our code, and the energy required to operate those servers and the mobile devices. We do not have control over how the devices and hardware that our application would run on are created and powered, but we do make sure to limit our GPS gathering so as to minimize the impact of our mobile application on the battery life of the mobile device.

### **13.9 Usability**

Our application is user friendly. As mentioned in our Testing section above, the employees at Broadband Holdings all reported that our product was intuitive and easy to use. There were no issues or confusion regarding how to perform a given task within our system.

### **13.10 Lifelong Learning**

This project inspired us to learn about technologies previously unfamiliar to us. We learned how to properly use Phonegap and how it interfaces with a Web server. We also had to take into consideration and research about the legality and ethics of tracking people while at work. This project also helped us learn how to manage ourselves and our time when dealing with a project with many facets of development.



## **13.11 Compassion**

Examining our application in the broader context of the developing world allowed us to consider how its features could be employed to promote safety. Many organizations operate to reach the most marginalized populations which are often logistically challenging to access. Reaching this market often puts the mobile workforce at risk of road traffic accidents which are responsible for the deaths of more than one million people annually [1]. If our application was deployed on feature phones, it would have the capability to act as a safety mechanism by keeping track of employees and to ensure that help reached them sooner should an accident occur.

## **14 Conclusions**

### **14.1 Summary of Work**

Throughout the course of our development, we reached almost all of our initial goals. We developed a mobile application that functions on both iOS and Android. It collects GPS location data of the device, and sends that data as well as a tracking ID and timestamp to a MySQL database for storage. The mobile application allows the user to choose when to start and stop tracking, as well as submit a check in form to provide more feedback. This check in information is also stored in the MySQL database.

We also created a Web interface to display this information. The Web interface has a secure login, along with all of the various webpages and functionality described above.

### **14.2 Lessons Learned**

Throughout the year of this project, we learned many lessons about software development and interacting with companies about a potential product.

#### **14.2.1 Developer Biases**

When developing software or an application, a developer's perspective traditionally has a bias, and that is why black box testing is so critical to the development process. We made critical mistakes when developing our application that we did not realize until we tested with potential users. For example, we did not consider testing our application in the background thoroughly to make sure it worked because we were only concerned with what the application did when it was open, not when it was in the background. Although an obvious requirement was that our mobile application should run in the background, we forgot to test it before we began black box testing.

#### **14.2.2 Benefits of Project Planning**

Our project was finished in time for Senior Design conference because of well planned project management throughout the year. The development timeline we drafted in the beginning kept us on track for the following months, and it was always a resource we could refer to in order to see if we were behind or on track. Additionally, from the beginning we realistically set our expectations for what our team of two could accomplish so that we would not suffer from scope creep, and fortunately, scope creep was not an issue for us. Realistic expectations, though, are normally not a reality in the business world, but

this lesson did teach us that it is important to negotiate with customers and managers to be clear what a developer team is entirely capable of in set period of time.

### **14.2.3 Limitations of Phonegap**

Phonegap made development a lot easier in the short amount of time we had, but it also presented some limitations to continuing our development when we came closer to completion. Phonegap is a great tool used to develop cross-platform mobile applications without having to worry about device-independent solutions. We realized that this solution is great for simple mobile applications, but as a mobile application grows more complex, there are a limited number of solutions that Phonegap provides. When we developed the background notifications portion of our application, we needed device-independent solutions for both iOS and Android devices. Phonegap did not have a all-in-one solution, unfortunately. To solve this issue in just the few weeks we had left, we researched and found a third party plugin that could be used for both iOS and Android device background notifications. However, this solution was temporary, and needs to be continuously updated with future versions of each platform - if the creator of the plugin does not update it himself. This example demonstrated just one way that Phonegap limits development of a growing, complex mobile application.

## **14.3 Suggested Changes and Future Work**

Through feedback gathered during our testing with Broadband Holdings, LLC, as well as being limited in time for completing this project, we have identified a few areas for changes, future work, and added functionality for our system.

### **14.3.1 Organize Aggregated Data**

First off, as mentioned in the Test Results section above, Broadband Holdings wanted to see more automated analysis of the data that we gather. In the devices page, a textual view of all the individual starting/ending tracking as well as check ins for a specific device are able to be seen. In addition, the user can filter that list of information by a specific date. We wish to display a specific date on a map visually, so a user can see a drawn route that a device traveled on a particular day. Another area for additional features would be with the data of the check ins. As it stands, our system allows a user to click on a check in for more details under the Activity Feed and Devices pages. However if a company has many check ins, it would be easier to have a dedicated page to displaying the check in information as well as performing some analysis on it, such as the amount of time that an employee spent on a certain job, or a particular type of job.

### **14.3.2 Integrated Devices and Manage Devices Page**

Having two separate pages for displaying information about a device (Manage Devices page) and displaying the location information gathered by a device (Devices Page) seemed to be rather redundant. To streamline this, we would like to integrate these two pages into one that is more of the style of the Devices page, along with an "Edit Device" button to allow for the functionality of the Manage Devices page. This would help to limit confusion about which page a user would need to navigate to to perform a specific task.

### **14.3.3 Feature Phone Testing**

We want to be able to eventually expand our application out of the realm of smartphones, and into markets where feature phones are more prevalent. Part of the initial motivation of this project was to provide a general purpose tracking application that could be used in a variety of areas, from a telecommunications company in the United States to rural health care workers in India. This is a longer term goal, as our system would need to be reworked to function and communicate through different mediums outside of typical cellular data or WiFi. An initial idea would be to transmit the location data via automated SMS messaging.

## 15 References

- [1] Annual Global Road Crash Statistics. *Association For Safe International Road Travel*. 2002. Web. 2 June 2016.
- [2] Chandra, Ankur, Shashank Jain, and Mohammed Abdul Qadeer. "GPS locator: An application for location tracking and sharing using GPS for Java enabled handhelds." Computational Intelligence and Communication Networks (CICN), 2011 International Conference on. IEEE, 2011.
- [3] Crosson, Stephanie. "Where's My Worker?: The Use of GPS Tracking Devices by Employers To Locate Employees." *Lawschoolblog.org*. 22 January 2014. Web. 10 January 2016.
- [4] Kaupins, Gundars, and Robert Minch. "Legal and ethical implications of employee location monitoring." System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on. IEEE, 2005.
- [5] Lee, Hyo-Haeng, In-Kwon Park, and Kwang-Seok Hong. "Design and implementation of a mobile devices-based real-time location tracking." Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on. IEEE, 2008.
- [6] Michael, Katina, Andrew McNamee, and M. G. Michael. "The emerging ethics of humancentric GPS tracking and monitoring." Mobile Business, 2006. ICMB'06. International Conference on. IEEE, 2006.
- [7] Moloo, Raj Kishen, and Varun Kumar Digumber. "Low-cost mobile GPS tracking solution." Business Computing and Global Informatization (BCGIN), 2011 International Conference on. IEEE, 2011.
- [8] Selvam, R., and V. Karthikeyani. "Mobile Software Testing- Automated Test Case Design Strategies." International Journal on Computer Science and Engineering 3.4 (2011): 1450-1461.
- [9] Tsui, Frank F., and Orlando Karam. Essentials of Software Engineering. Sudbury, MA: Jones and Bartlett, 2007. Print.

## 16 Appendices

### 16.1 Installation Guide

#### 16.1.1 Environment Requirements

In order for both the website and mobile application to work properly, you must already have a web server to host the PHP scripts and website files, and a MySQL database to hold the data for the application. Additionally, in order to compile and download the Phonegap application, you must have Phonegap or Cordova downloaded and working properly either through the command line or desktop service. More details can be found here, <http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/>. In order to download the Phonegap application onto a mobile application instead of running it in an emulator, Phonegap has device-specific requirements. For running the application on an iOS device, an Apple Developer Account is required and the latest version of Xcode must be installed. For more information about platform guides, you can turn to the Cordova developer docs for requirements <https://cordova.apache.org/docs/en/3.0.0/guide/platforms/>.

Once Phonegap is set up, a third party plugin needs to be installed to allow the application to run in the background. Depending on whether you are running Phonegap or Cordova, run the following command from the command line within the phonegap directory to install the plugin:

```
cordova plugin add https://github.com/pmwisdom/cordova-background-geolocation-services.git
```

or

```
phonegap plugin add https://github.com/pmwisdom/cordova-background-geolocation-services.git
```

More information about how to use the plugin can be found at the following url <https://github.com/pmwisdom/cordova-background-geolocation-services>.

#### 16.1.2 Database

It is recommended that phpmyadmin be used to set up and maintain the database tables. Database tables are needed for the mobile application and for the website. Once these are created in MySQL, then the database has been set up.

- The stw users table stores the website users, a username and password need to be manually added to this table.
- The st checkin table stores all the check in data that is submitted by a mobile application.
- The st current locations table displays the locations of all users who are currently being tracked.
- The st devices table displays all the mobile devices that registered with the system
- The st recent activity has all the activity feed updates and if the activity is a check in, then a specific check in id is stored in the table so that details can be matched with the update.

The following tables should be created in MySQL:

---

```

CREATE TABLE 'stw_users' (
  'user_id' int(11) NOT NULL AUTO_INCREMENT,
  'username' varchar(30) DEFAULT NULL,
  'fname' varchar(30) DEFAULT NULL,
  'lname' varchar(30) DEFAULT NULL,
  'date_joined' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  'passwd' varchar(30) NOT NULL,
  PRIMARY KEY ('user_id')
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1

CREATE TABLE 'st_checkin' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'tracking_id' int(11) NOT NULL,
  'client_name' varchar(80) NOT NULL,
  'work_order' int(11) NOT NULL,
  'time_started' time NOT NULL,
  'time_ended' time NOT NULL,
  'feedback' varchar(400) NOT NULL,
  'timestamp' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=78 DEFAULT CHARSET=latin1

CREATE TABLE 'st_current_locations' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'tracking_ID' int(11) NOT NULL,
  'name' varchar(30) NOT NULL,
  'lat' float(10,6) NOT NULL,
  'lng' float(10,6) NOT NULL,
  'address' varchar(80) NOT NULL,
  'timestamp' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=1135 DEFAULT CHARSET=latin1

CREATE TABLE 'st_devices' (
  'device_id' int(11) NOT NULL AUTO_INCREMENT,
  'tracking_id' int(11) NOT NULL,
  'fname' varchar(50) NOT NULL,
  'lname' varchar(50) NOT NULL,
  'company' varchar(30) NOT NULL,
  'title' varchar(80) NOT NULL,
  'phone_number' varchar(15) NOT NULL,
  'email' varchar(60) NOT NULL,
  'last_seen' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
  PRIMARY KEY ('device_id')
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1

CREATE TABLE 'st_location_data' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'tracking_id' int(11) NOT NULL,

```

```

'name' varchar(30) NOT NULL,
'lat' float(10,6) NOT NULL,
'lng' float(10,6) NOT NULL,
'address' varchar(80) NOT NULL,
'timestamp' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=685 DEFAULT CHARSET=latin1

CREATE TABLE 'st_recent_activity' (
'id' int(11) NOT NULL AUTO_INCREMENT,
'checkin_id' int(11) NOT NULL,
'type' varchar(25) DEFAULT NULL,
'tracking_id' int(11) NOT NULL,
'description' varchar(250) NOT NULL,
'timestamp' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=615 DEFAULT CHARSET=latin1

```

---

### 16.1.3 Website

The website is the simplest portion to install. All the files under `www` should be moved to a server. Once the files have been moved, the database credentials need to be changed under the file `config.php`. The `$host_name` variable should be assigned to your server's hostname and the MySQL credentials should be updated. Once this is done, the database and website should be working properly as long as your server has the global read permissions set.

### 16.1.4 Mobile Application

In order to run the mobile application, the currently installed version of Phonegap must support whatever platform the application is run on. This step should be completed in the Phonegap installation process, but if not, you can refer back to the developer guide mentioned above.

Installation of the mobile application is quite simple for both Android and iOS devices. If the Phonegap project has already been compiled, then a `.apk` file should exist in `/platforms/android/ant-build`. For iOS platforms, the xcode project file can be found under `/platforms/ios/ProjectName.xcderoj`. Both of these files are ready to run, iOS devices however require some specific configurations mentioned in the next section.

### 16.1.5 iOS Special Configurations

For our iOS project to be able to run in the background and collection GPS and location updates, the `{Project Name}.plist` file for the iOS app must be edited to enable certain permissions. At the end of the `{Project Name}.plist` file, right before the `</dict>` tag, add the following code to enable location services, GPS, background mode permissions, and to allow the application to receive external requests

---

```

//allow external requests CDNs

```

```

<key>AllowsArbitraryLoads</key>
  <true/>
<key>NSAppTransportSecurity</key>
  <dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
  </dict>
<key>NSLocationAlwaysUsageDescription</key>
  <string>${EXECUTABLE_NAME} Would Like to Use Your Current Location Even In
    Background.</string>
//allow device to retrieve gps and location-services
<key>UIRequiredDeviceCapabilities</key>
  <array>
    <string>location-services</string>
    <string>gps</string>
  </array>
//allow app to run in the background
<key>UIBackgroundModes</key>
  <array>
    <string>location</string>
    <string>fetch</string>
  </array>

```

---

## 16.2 User Manual

This user manual is intended to provide users with instructions on how to use both the Web Interface and Mobile Application portions of SaguaroTrack.

### 16.2.1 Administrator Web Interface

#### 1. Login

- After navigating to the URL of the website as given by your company, enter in your username and password into the appropriate fields and click Login.
- If your login information is correct, you will be admitted to the system. Otherwise, you will be prompted to reenter your credentials.

#### 2. Activity Feed

- The Activity Feed shows the most recent events of the system. It shows when users start tracking, end tracking, or check in.
- To view more details of one of the check ins, click the "Click for details" link below the entry title.
- To view more than just the most recent five entries, click the load more button at the bottom of the page.

#### 3. Map

- Select the Map tab at the top of the page to view a real time map of all the devices currently being tracked



- On the left, there is a panel with a list of names along with a location associated with that name. These are all the people who are actively being tracked. The search bar can be used to search the name of a particular employee.
- On the right, there is a map displayed with markers for the locations of each of the employee. Click on a marker to display the name of the employee it belongs to.
- The map can also be searched by a location to see any employees who are near that area.

#### 4. Devices

- Select the Devices tab at the top of the page to see a list of all the devices in the system along with an individual tracking history for each device.
- On the left, the various users in the system are listed. Click on a user in the left panel to bring up more information about them.
- After clicking on a user, more information about them will display on the right. The Recent Activity is a scrollable section that functions essentially the same way as the Activity Feed page, but just with one device. To see more information about the check ins, again click on the "Click for details link."
- To filter the individual's activity by a particular day, click in the select date box. A date selector will pop up, and you can click on a particular day. The recent activity list will automatically update to reflect just the tracking events for that user on the selected day. To reset the recent activity list to its default, simply click the reset button.

#### 5. Manage Devices

- Select the Manage Devices tab at the top of the page. You will be prompted to enter authentication to access this webpage as it allows a user to edit information about the devices in the system.
- After entering in your authentication credentials, there will again be a list of the devices in the system. Click on a device to see and edit the information associated with it.
- Any of the fields can be changed as desired. To save the changes, simply click the save changes button.
- To remove a device from the system, click the "Remove Device" button, and click ok to the prompt that pops up making sure that you want to delete that device. Refresh the page to see this change reflected.
- To add a device to the system, click the "Add Device" button, fill out the fields, and then click submit. If a user already has the tracking ID that was assigned, you will be notified and the new user will not be added.

### 16.2.2 Mobile Application

#### 1. Login

- To login to the system, enter your tracking ID given to you by your supervisor. If this is a correct ID, you will be admitted into the application.

## 2. Start Tracking

- To start tracking, simply push the "Start Tracking" button at the top of the page. The screen will change to show a marker on the map of where you are, and show textually your location as well.
- If you wish to use another application on your phone, simply minimize the application, and it will continue to gather your location data. Each time the data is gathered, you will receive a notification as a reminder that you are still being tracked by the system.

## 3. End Tracking

- To end tracking, simply push the "End Tracking" button at any point. All location gathering will be stopped, and you will be returned to the same home screen just after logging in.

## 4. Check In

- To check in and record feedback about a particular job site, simply push the "Check In" button at the bottom of the screen, below the map.
- A new page will display with various fields to fill out. Fill out the fields as labeled, and press "Submit." After this, your information will be sent to the server, and you will be taken back to the tracking screen.
- If you do not wish to submit the feedback at this time, simply push "Cancel" at the bottom of the screen to be taken back to the tracking screen.

## 5. Logout

- To properly logout of the application, first push the "End Tracking" button, and then push the "Logout" button. You will be returned to the initial login screen with the prompt to enter in your tracking ID.