**Silent Technical Alert Response Team**

**(START)**

Cohner Marker

Spencer Johnson

Jonathan Barella

**Android GUI Implementation and Basic Use**

**Concept**

Our project idea originated from the problem of small companies not being able to afford an Alert System. Many of the big-time Alert-System companies require a specific minimum amount of people, usually around 500, and companies that only have 20 or so employees do not have a reasonable, easy way to get an alert system. So our Application is designed to help these small businesses feel more secure without having to deal with giant prices or not being able to enter the market at all for being to small.

**Execution**

Our Application, currently Android based, will have a simple but fast user interface where a person in trouble can quickly send out a silent emergency message. Then the response server will be able to listen for all incoming messages and make an appropriate response. For example, say a person with a gun were to enter the building. A person with our application would be able to, in a few easy touches, send out a detailed enough message to the sever that the police, SWAT team, or other administrators can respond immediately with enough information to save as many lives as possible. Time and information are important aspects between life or death, and we hope our application give faster better communication between people who are in trouble and people who can help

**Advanced Features**

* ***Location Features***
  + *GPS User Location:*  One of the most important thing is about keeping the user of the application will be knowing exactly where the location of the person is**.** It is impossible to save anyone when their location is unknown. Upon first activation of the app, it immediately send the current location of the person via the inbound Android GPS services, allowing for the emergency services to know precise location.
  + *Advanced User Locations:* Once the exact GPS coordinates of the user are known and the proper emergency services are notified, the app will attempt to learn as much information about the location as possible. If the user is safe or able to answer, it will ask for information about the location, such as where the entrances are, what floor the user is on, and where the danger is.
  + *Authority GPS Locations:*With proper integration with the police systems, active tracking on the location of the emergency vehicle will be viewable in app. Whether the app will display a map or simply will show how far away the services are, it is important that the user is updated on how long it will take and have adequate information regarding the situation
* **User Interface**
  + *Fragments:* As any app that is focused on saving lives, the usability and speed of the app must be top priorities. START uses linked fragments that can swap back and forth very quickly with no loading times nor excessive animations. Information must be sent quickly and not bogged down with over the top flashy screens, which would make it much harder to read in a stressful situation.
  + *Button Transition:* The apps initial questions will be for the most important things that the police will need to know, such as:
    - GPS coordinates
    - If the user is safe or still in a dangerous situation
    - The nature of the emergency
    - Which emergency services to contact (Police, ambulance, swat..)
    - Whether or not they are able to be contacted in other ways, such as by phone

The app will display a list of options to get information as quick as possible, without having to fill out several forms or having to type. The buttons can be moved through quickly and are chosen simply by the user tapping on the option. After the most important information is passed, it will inquire further if possible, about things such as

* Location and number of entrances to building
* Number of floors in the building and which floor the user is on
* Number of dangerous objects (Number of Shooters, bombs, poison sources etc.)
  + *Flexibility:* One of the less obvious features is the ease of changing the app. It is easy to add more fragments or buttons without much changing of previous code. This allows for the addition of more questions with ease. As the app improves, more specific options will be added with more specific questions for each emergency department notified.
* **Security and Backup**
  + *Prevention of Accidental Alert:* As this app does notify emergency services such as 911, accidental use of the app or unintended use by others are problems. A few solutions would be
    - Only allowing the app to be used during certain times, which the user would set to be his or her work hours, after hours would require the entering of a pin number in order to access the apps services.
    - Location settings that will only allow the app to be used when you are within a certain radius of the user’s work location. If outside the boundaries of the app, it would require a pin to use. Battery life would be a problem with this method.
    - Constant use of a pin. Any time the initial help request is sent, the app would request a PIN. This would entirely eliminate accidental calls, but would slow down the process of sending the alert.

Each Solution has its advantages and disadvantages so the final design of this aspect has not been fully decided.

Apache (PHP) Server Implementation

# Proposed Infrastructure

## Problem

Many small businesses have their own homemade infrastructure that is set up much like the diagram shown on the next page. A lot of current companies will charge amounts that are well out of the rage of the small business for establishing an alert system in their home solution. Also, a lot of alert systems are not even hosted on site but instead in the company’s own datacenter. This can lead to more complications and a lot of money paid out to the companies.

## Current Solutions

* + On Site Hosting
    - Alert Companies charge insane amounts of money to implement their own services in the small business infrastructure. In order for them to do this, the small business must have a certain number of employees to even apply.
  + Remote Hosting
    - Alert Companies host the alert system in their own datacenter. However, this can lead to complications and the need of a full staff, response on phone, support etc. This leads to expense monthly charges and also they do not guarantee fast emergency responses as they could be hosted in a different state entirely.

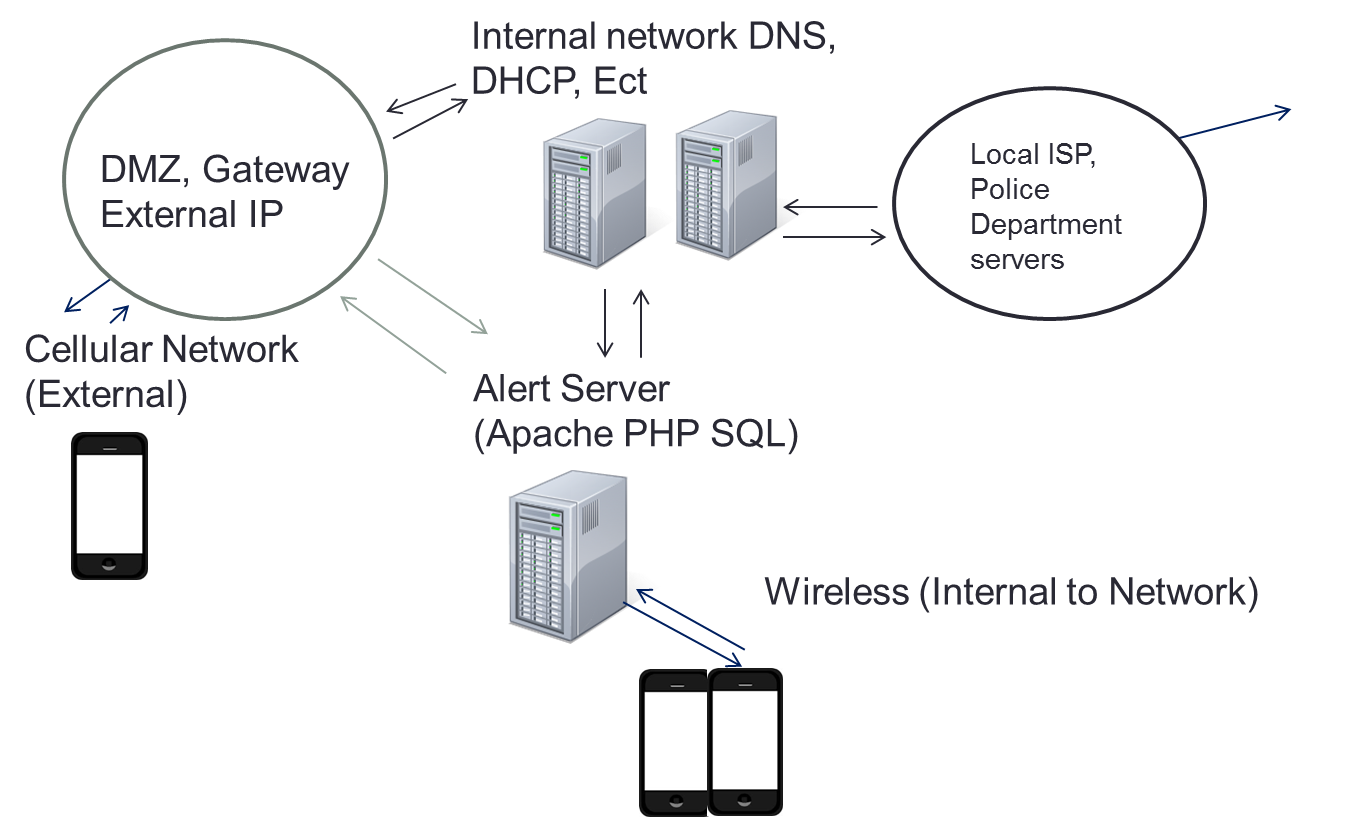
## Proposed solution

This proposed server infrastructure will be hosted in the internal network of the small business, but unlike the current solutions will be affordable and easy to implement. It will not require a certain threshold of people to apply for it. This will allow the small companies to be able to manage the alert system in their own way (but still ensure functionality), and still guarantee the fast responsiveness and of having it on site. The service will be ready at any point in time and can respond to responses whenever an authorized request comes in. This infrastructure hopes to allow small businesses to have a simple, fast, and affordable solution.

# Users/Administrators of Apache PHP Server

* Employees
  + Interface to the server through the mobile application
  + Receive Lockdown Responses
* Response Administrators
  + Can edit/add/update Employee information
  + Receive detailed information on emergency
* Local Emergency Response Authority
  + Will receive a message containing the needed information
  + They will choose the course of action they were trained on
* Server Administrators
  + Will be able to update/change SQL tables, maintenance on the server, and choose the local emergency response authority.

# Basic Concept



# Features

## Apache PHP Server Features

* + Service
    - Server will be able to implement basic PHP and SQL functionality
    - This server will provide a constant “resting” service that will respond to requests immediately.
  + Interface with Android
    - In order to interface with android, the server will be accepting HTTP POST requests and other forms of requests from the user’s android phone. This will allow the server to receive messages from both internal and external requests.
  + Interface with Emergency Authority
    - The server will be able to send a message/alert to the community 911 emergency response, or other chosen emergency authority, with all of the information obtained from the alert message of the android app. Also, all of the designated response administrators of the application will be contacted. Essentially, this means if the small business has designated security personnel, the app will contact that person immediately of the emergency in addition to the 911 emergency responses.
  + Lockdown message to Employees in Danger
    - After an employee has signified a danger in the work place, all other employees will receive a calm, detailed message on how to properly respond to the situation. For example, if an employee sends an emergency response of a live shooter, all employees will be notified to of the proper response of locking their office, or any other doors and entering a lockdown.
  + Implement Login Functionality
    - In order to improve security, identification of false reports, and fast responses we are looking at a pre-authorize option. When a phone connects to the server, via internal network or external (cellular), it will log in to the apache server through a check with a database. If the user is indeed authorized, every time that person comes to work he/she will be not have to log in everyday but instead will be ready to go from day to day. This solution would be able to manage the employees outside of the internal network (Cellular) if we implement a log-in setting option where the user can login via the GUI of the phone and the phone can use that info on each time the app is started.
  + Disclaimer
    - Un-authorized phones, tablets, and laptops should not be able to send an emergency message. However, this response system is not designed to be hard to use and take five or more minutes to formulate a response. Therefore we would like to make this response system semi-secure to deter bots and basic attacks. If a false report has been filed the user and phone from the false report will be investigated by the local law enforcement and subject to the full extent of the law, as calling in false emergency responses is highly illegal.

## SQL Server Features (Subject to needs of Small Business)

* + Tables
    - EmployeeLogin
      * EmployeeUserName
      * EmployeePassword
    - ResponseAdminLogin
      * ResponseAdminUserName
      * ResponseAdminPassword
    - Employee Information (Used for Response Messages, Subject to expansion)
      * EmployeeUserName
      * EmployeeOfficeNumber
      * EmployeeCellularNumber
      * EmployeeStatus
    - EmergancyContact
      * EmergancyContactName
      * EmergancyContactPhone
      * EmergancyContactEmail
      * EmergancyContactLocation

## DNS, DHCP, Basic Infrastructure

* + The basic Infrastructure of a business (Mail, DNS, DHCP, and Networking) should already be implemented in the work place. Most commonly, these are homemade systems that have been in place with the business for a long time.
  + In the event of the small business not having a proper set up, we can refer them to a designated IT counseling or other service to help get the business up and running for the proper infrastructure needed for a response system.

**Use Case: *ASP.NET***

**CHARACTERISTIC INFORMATION**

**Goal in Context:** *The website implementation of the START service will allow users who do not have access to an Android device the ability to use the service. The website will also provide a front end user interface for manipulation of the user data.*

**Scope:** *This implementation will use the Model View Controller design pattern implementation within the ASP.NET framework with the primary language being C#. The front end will use the LINQ and TSQL languages to manipulate the backend database.*

**Level:** *one of two:* Account Manipulation, Event Reporting, View Past Events

**Preconditions:** *The Linux Server must already be established for reporting to work correctly.*

**Success End Condition:** *Small businesses and users without devices running android will be able to access START services. A simple web interface will provide simple access to the database to view records and manipulate current users*

**Failed End Condition:** *Small businesses and users without android devices will not be able to use START services. There also will be no simple interface for the related business data. Administrators will have to directly access the database to view and manipulate the databases contents.*

**Primary Actor:**

***Reporting Events:*** *All employees of the organization implementing START Services*

***Account Manipulation:***  *Small business owners, and administrators*

***Viewing Past Events:*** *Small business owners, organizational leadership, administrators, Law Enforcement Officials*

**Trigger:** *User accessing the website from a capable device*

**MAIN SUCCESS SCENARIO**

***General Setup:***

1. *Administrator creates user account*
2. *User granted permissions for data manipulation, reporting*
3. *User personal information updated to include:*
   1. *First Name*
   2. *Last Name*
   3. *Contact Phone Number*
   4. *Email*
   5. *Job Code (at discretion of organization)*
   6. *Physical Address*
   7. *Location in Building*

***Reporting:***

1. Web page loaded cosc4210.cs.uwyo.edu/students/START
2. User Login (Future SSO location)
3. Single Web Form Loaded(multi frame)
   1. Help, Big Button( Frame 0, Auto Post Back (APB) = True )
   2. Are You Safe? Yes/No (Frame 1, APB = True)
   3. What is the nature of your emergency? (Frame 2, Active shooter, fire, bomb, medical situation, APB = True))
   4. If Safe, Describe your situation(Frame 3, text Box, Submit Button)
   5. Send Relevant information to Linux PHP server to include:
      1. Profile Information
      2. User Safety
      3. Event Type
      4. Description (optional)
   6. Load Google.com to disguise event report

***Account/Data Manipulation***

1. Login to Hidden Administration page with admin account
2. Brought to Data portal with manipulation links
3. Manipulate desired data to include:
   1. Creation of user accounts
   2. Modification of User Account data
   3. View Web reported events

**RELATED INFORMATION**

**Priority:** *High on par with implementation of START services on Android*

**Performance Target:** *Project Completion Date*

**Frequency:** *As often as needed by the implementing small business or organization*

**Superordinate Use Case:** *PHP server*

**Subordinate Use Cases:** *optional, depending on tools, links to sub use cases*

**Channel to primary actor:** *web server hosted on csweb.cs.uwyo.edu, local database or external database (potentially CSTFS)*

**OPEN ISSUES**

*Implementation of a single sign on code (SSO) that would enable event reporting bypassing the standard username/password login*

*How many past events and what details should be immediately available?*