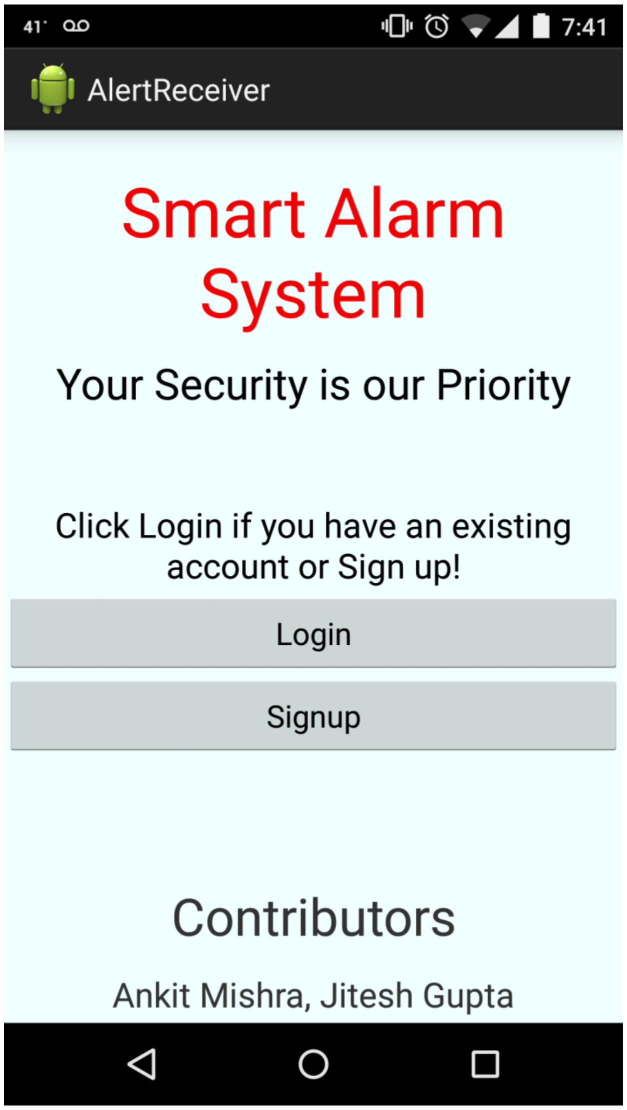
CIS 542 - Final Project
Smart Alarm Sensor

Ankit Mishra, Jitesh Gupta

Brief -

The project is implemented as an android application along with a self-hosted web server. The purpose of the project is to create a sensor system that allows the user to remotely monitor and control sensors. This is facilitated by providing an android application that can run as a sensor or a receiver/monitor. The user needs to deploy the application on an android phone and use it as a sensor. The sensor application will then read light, sound and motion sensor readings and upload them on a server. The user can then install the android application on another phone and use it as a receiver to remotely monitor the sensor reading. The user also has the control of turning on an alert mode from either of the devices. When turned on, the user will receive alert notification whenever the sensor readings cross certain thresholds.



Smart Alarm System

Your Security is our Priority

Click Login if you have an existing account or Sign up!

Contributors

Ankit Mishra, Jitesh Gupta

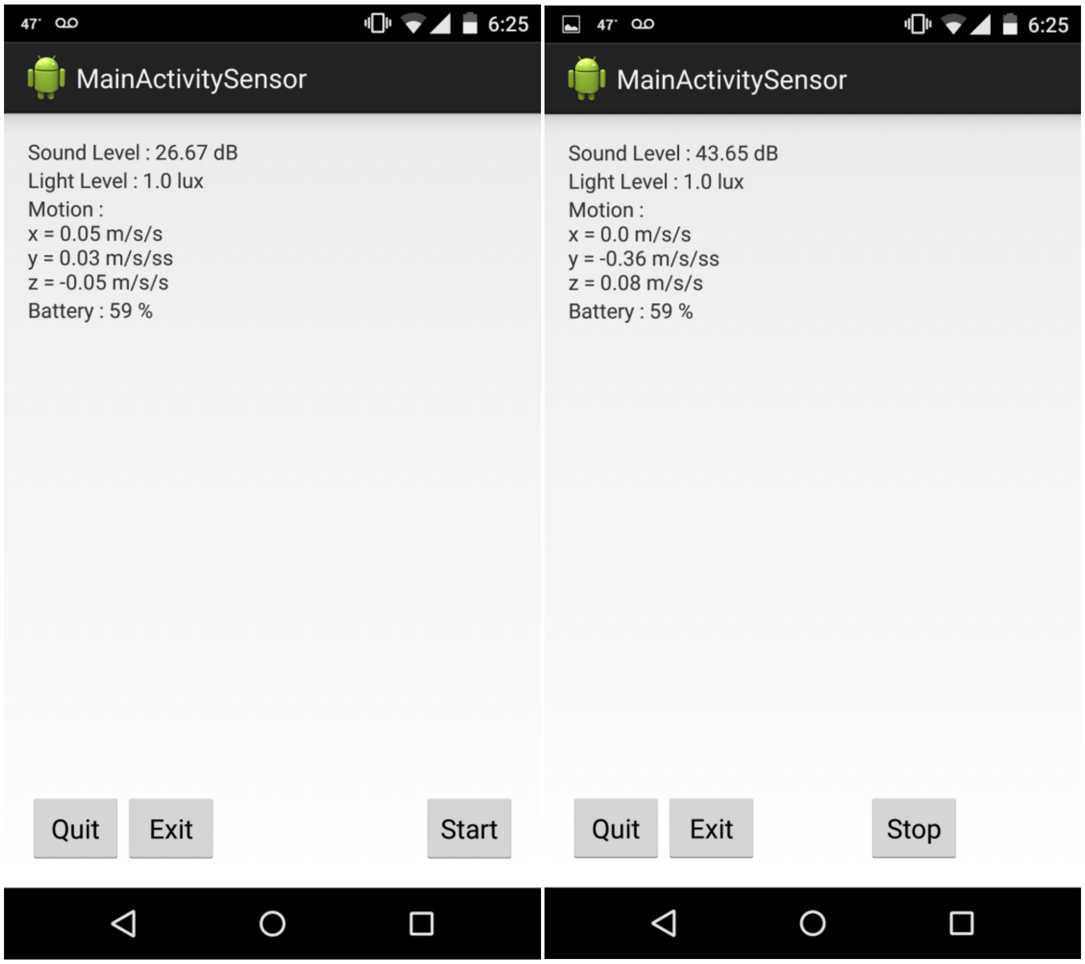
CIS 542 - Final Project
Smart Alarm Sensor

Ankit Mishra, Jitesh Gupta

System Components -

The system includes three main components -

1. **Sensor -**
   The sensor is an android phone which has the smart alarm sensor application installed. The application reads the sensors present in the android phone and continuously uploads those readings onto a web server. The sensor has the feature of turning on the alert mode of the system using buttons provided in the user interface of the application.



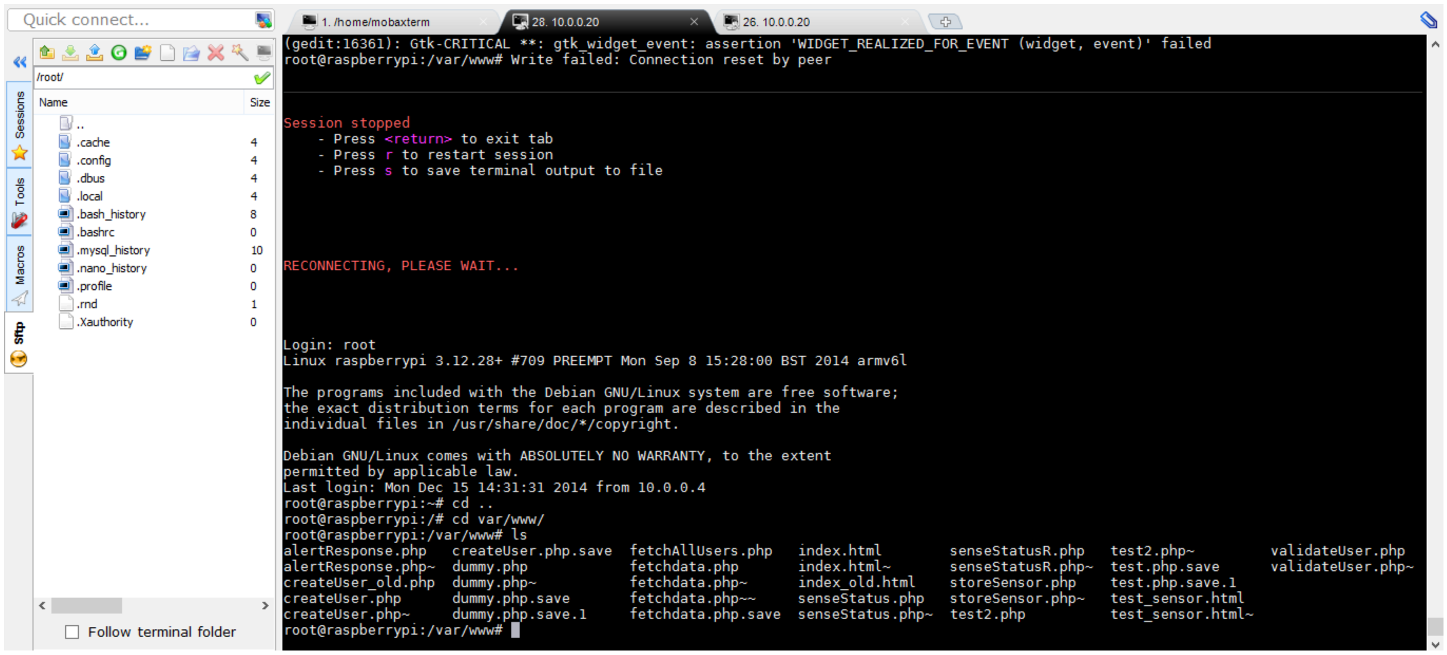
Sound Level : 26.67 dB
Light Level : 1.0 lux
Motion :
x = 0.05 m/s/s
y = 0.03 m/s/ss
z = -0.05 m/s/s
Battery : 59 %

Sound Level : 43.65 dB
Light Level : 1.0 lux
Motion :
x = 0.0 m/s/s
y = -0.36 m/s/ss
z = 0.08 m/s/s
Battery : 59 %

CIS 542 - Final Project
Smart Alarm Sensor
Ankit Mishra, Jitesh Gupta

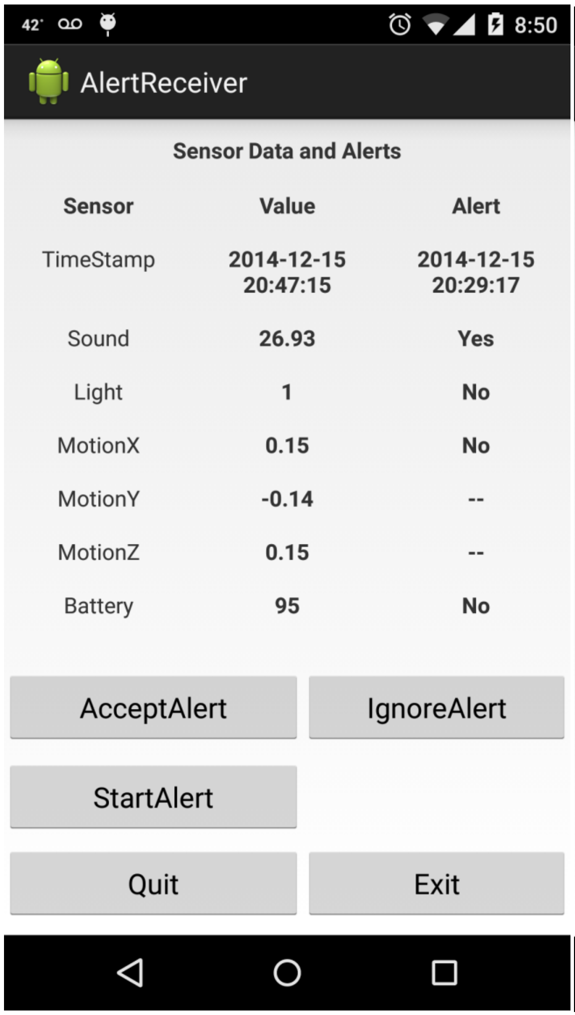
2. Web Server -

The web server is mainly an apache server hosted on a Raspberry Pi development board. The web server involves an FTP and a MySQL system as well. The web server responds to web requests by the sensor and the receiver via php files. For each request the sensor/receiver sends an HTTP POST request to a php file within the server. Each php file is designed to read the data within the HTTP request and use the data to query the MySQL database for the requested information. The requested information is then encoded within a JSON object and returned to requesting application. The server has php files designed for operations like - storing the sensor readings into the database (storeSensor.php), reading the sensor readings from the database (fetchdata.php), among others. The server is connected to the internet via a free domain at **mishra14.ddns.net.**



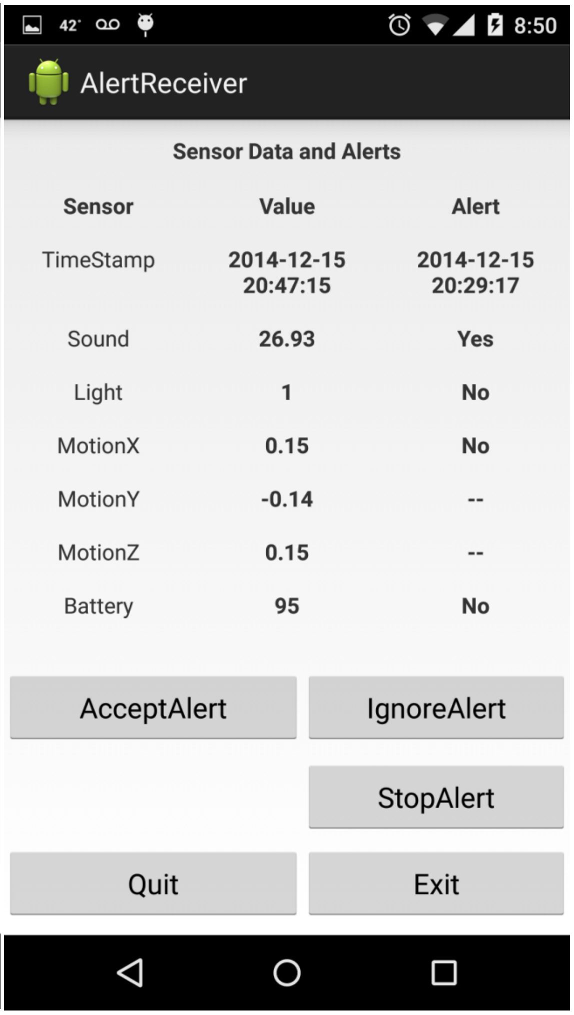
```
(gedit:16361): Gtk-CRITICAL **: gtk_widget_event: assertion 'WIDGET_REALIZED_FOR_EVENT (widget, event)' failed
root@raspberrypi:/var/www# Write failed: Connection reset by peer


Session stopped
    - Press <return> to exit tab
    - Press r to restart session
    - Press s to save terminal output to file




RECONNECTING, PLEASE WAIT...




Login: root
Linux raspberrypi 3.12.28+ #709 PREEMPT Mon Sep 8 15:28:00 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 15 14:31:31 2014 from 10.0.0.4
root@raspberrypi:~# cd ..
root@raspberrypi:/# cd var/www/
root@raspberrypi:/var/www# ls
alertResponse.php   createUser.php.save  fetchAllUsers.php   index.html         senseStatusR.php   test2.php~         validateUser.php
alertResponse.php~  dummy.php            fetchdata.php       index.html~        senseStatusR.php~  test.php.save      validateUser.php~
createUser_old.php  dummy.php~           fetchdata.php~      index_old.html     storeSensor.php    test.php.save.1
createUser.php      dummy.php.save       fetchdata.php~~     senseStatus.php    storeSensor.php~   test_sensor.html
createUser.php~     dummy.php.save.1     fetchdata.php.save  senseStatus.php~   test2.php          test_sensor.html~
root@raspberrypi:/var/www# 
```

3. **Receiver -**
The receiver is implemented in the form of an android phone which is running the smart alarm sensor application in the receiver mode. The receiver has the main task of requesting the sensor data from the web server, which was stored by the sensor. Further the receiver also requests the web server for any active alert information and displays all of the information on the screen. The user interface is designed to allow the user to toggle the Alert Mode which is the synchronized with the web server and reflects on the sensor application as well. Further the user can also accept or ignore any active alert.



Sensor Data and Alerts

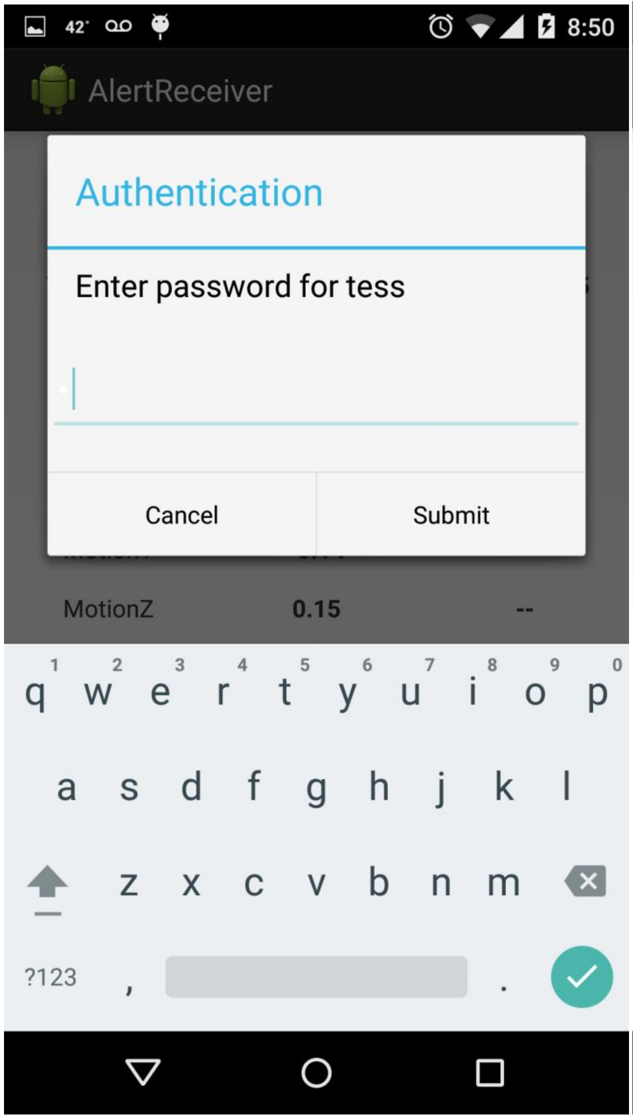
Sensor	Value	Alert
TimeStamp	2014-12-15 20:47:15	2014-12-15 20:29:17
Sound	26.93	Yes
Light	1	No
MotionX	0.15	No
MotionY	-0.14	--
MotionZ	0.15	--
Battery	95	No

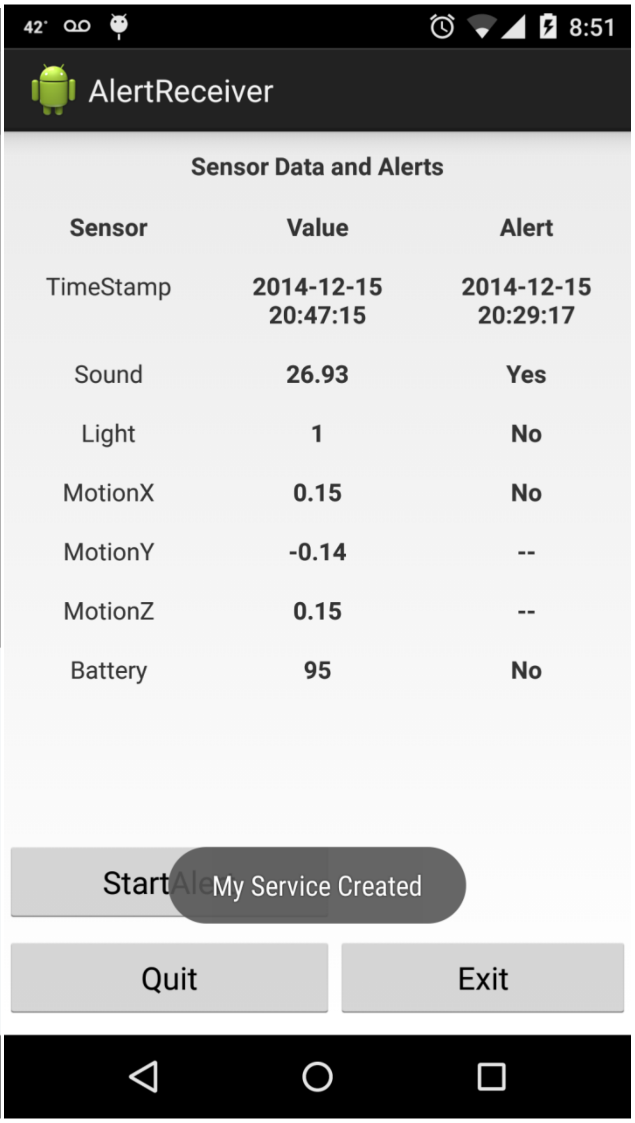


Sensor Data and Alerts

Sensor	Value	Alert
TimeStamp	2014-12-15 20:47:15	2014-12-15 20:29:17
Sound	26.93	Yes
Light	1	No
MotionX	0.15	No
MotionY	-0.14	--
MotionZ	0.15	--
Battery	95	No

CIS 542 - Final Project
Smart Alarm Sensor
Ankit Mishra, Jitesh Gupta



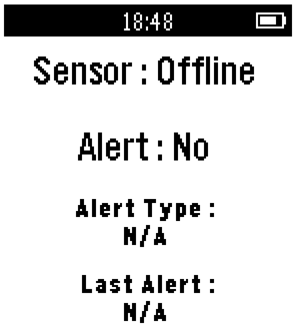


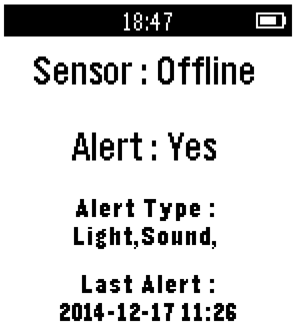
Sensor Data and Alerts

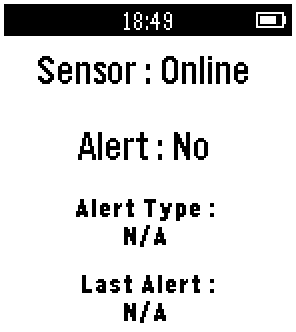
Sensor	Value	Alert
TimeStamp	2014-12-15 20:47:15	2014-12-15 20:29:17
Sound	26.93	Yes
Light	1	No
MotionX	0.15	No
MotionY	-0.14	--
MotionZ	0.15	--
Battery	95	No

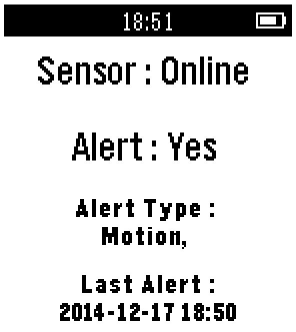
4. Pebble Watch -

The system also contains a pebble watch application which can also be used as an auxiliary receiver. The pebble watch application is independent and stand alone application which is designed to fetch the status of the sensor and any previous open alerts for the user. The user can also configure the application to send alert emails everytime a new type of alert is generated from the web server. The pebble watch enables the user to quickly check if the sensor is online/offline and also look at the recent alert. This reduces the need to open and check the receiver side of the smart alarm system application of the phone and allows ease of use for the user.


18:48
Sensor : Offline
Alert : No
Alert Type :
N/A
Last Alert :
N/A


18:47
Sensor : Offline
Alert : Yes
Alert Type :
Light,Sound,
Last Alert :
2014-12-17 11:26


18:49
Sensor : Online
Alert : No
Alert Type :
N/A
Last Alert :
N/A


18:51
Sensor : Online
Alert : Yes
Alert Type :
Motion,
Last Alert :
2014-12-17 18:50

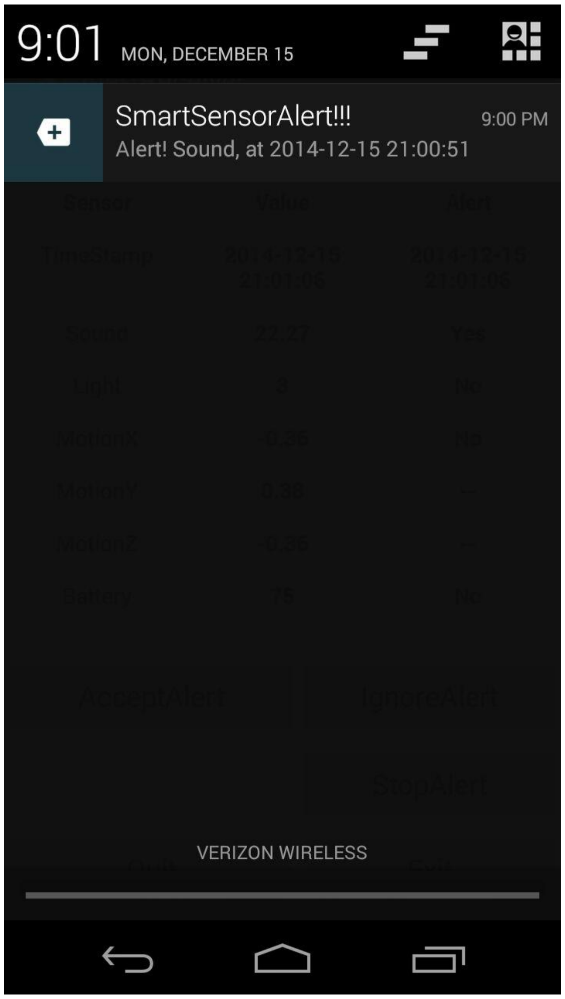
CIS 542 - Final Project
Smart Alarm Sensor

Ankit Mishra, Jitesh Gupta

System Features -

1. **Remote Sensing -**
   The system facilitates the user to remotely read sensor readings and monitor the sensor status.

2. **Alert Mode -**
   The user can toggle the Alert mode from either of the two applications which then reflects on the other application.
   When in alert mode alerts are generated to inform the user when the sensor values cross the thresholds.

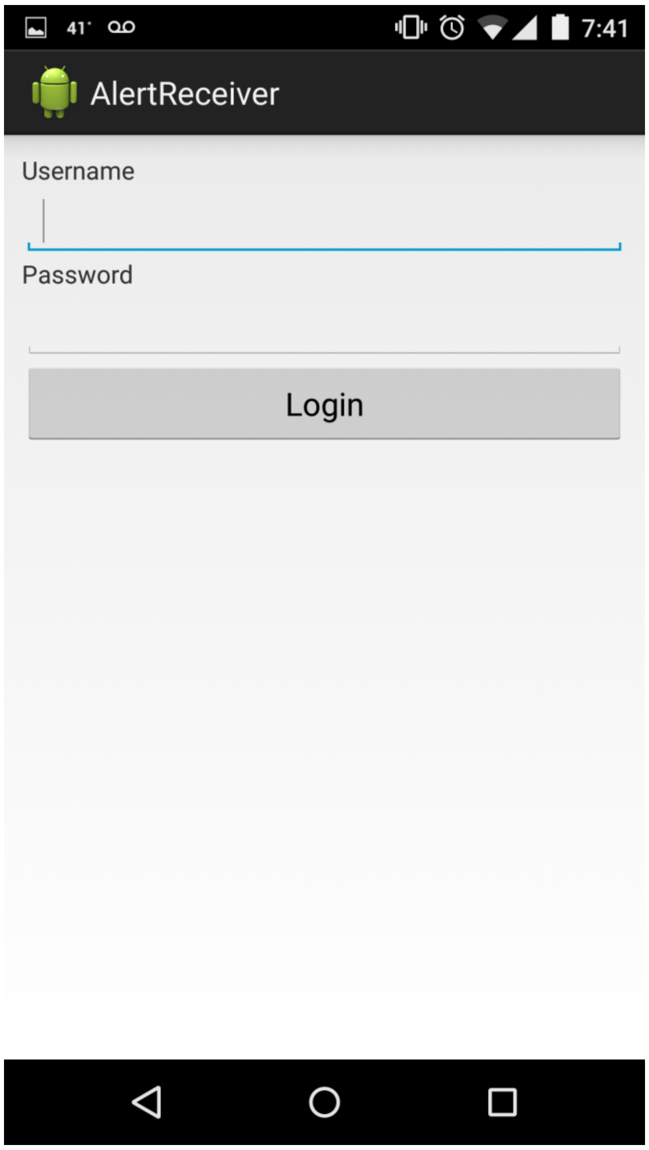

9:01 MON, DECEMBER 15

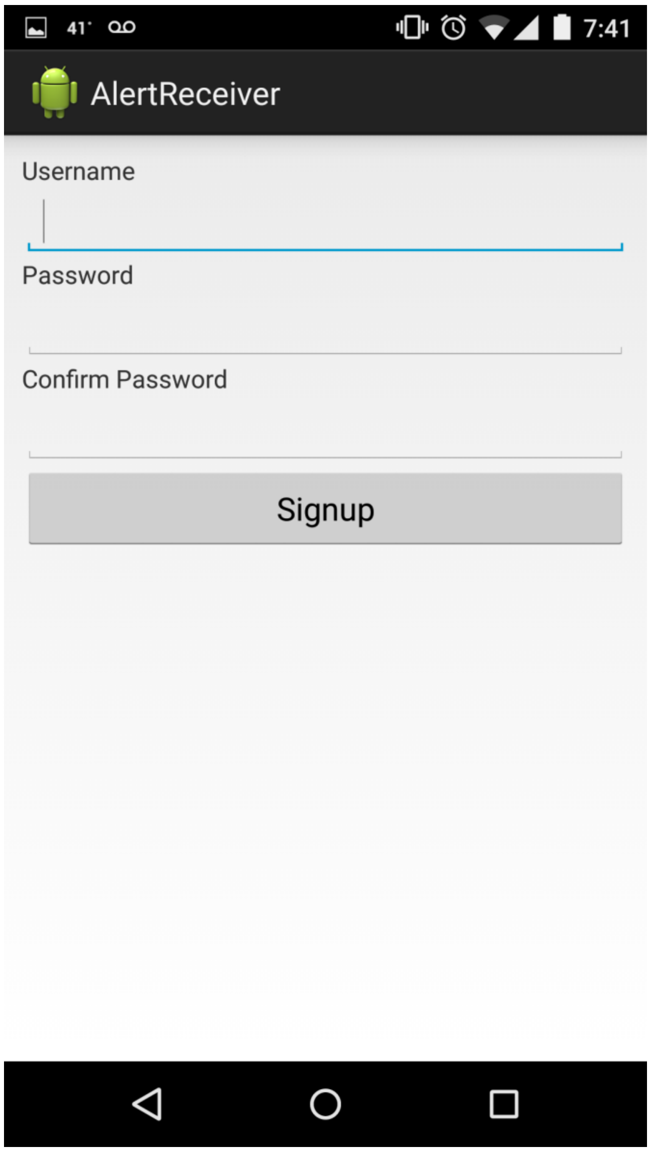
SmartSensorAlert!!! 9:00 PM
Alert! Sound, at 2014-12-15 21:00:51

Sensor	Value	Alert
TimeStamp	2014-12-15 21:01:06	2014-12-15 21:01:06
Sound	22.27	Yes
Light	3	No
MotionX	-0.36	No
MotionY	0.38	--
MotionZ	-0.36	--
Battery	75	No

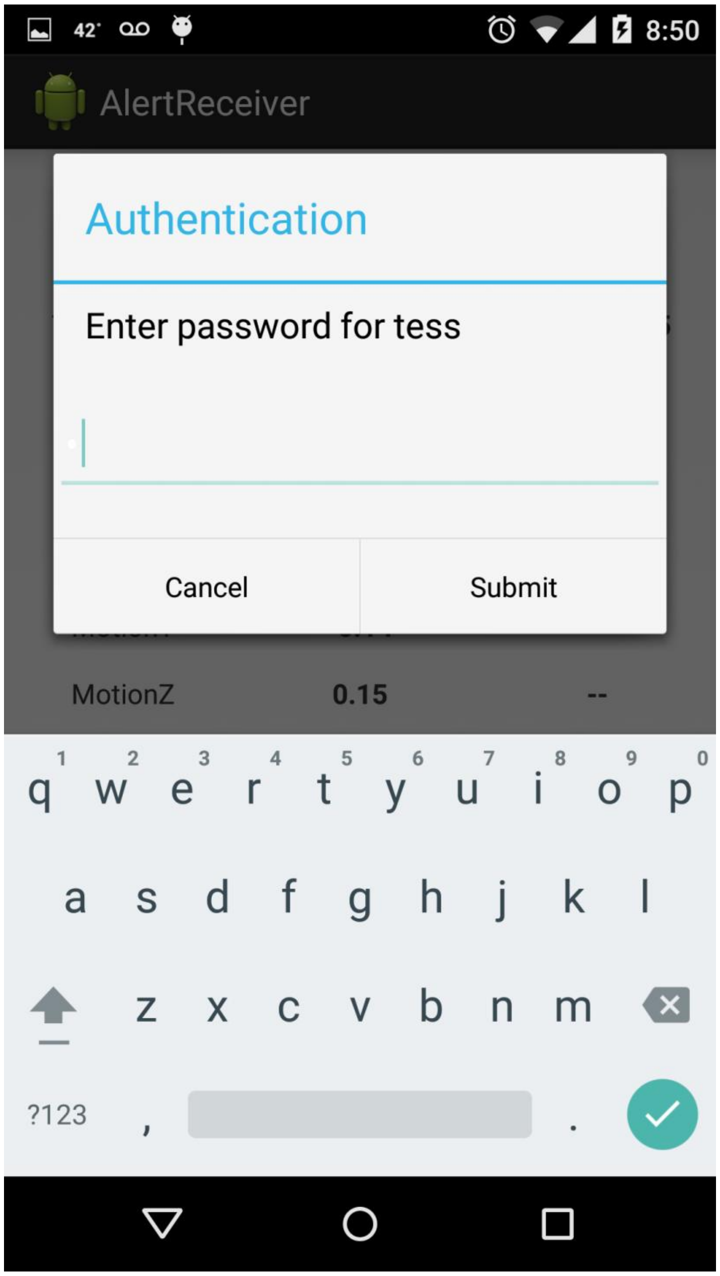
3. **Dynamic Thresholds -**
   The user has control over the thresholds. The thresholds are set based on the current sensor readings when the Alert Mode is turned on. The user can further adjust the thresholds by dismissing an alert. Each time the user dismisses an alert, the thresholds are updated in the database to reflect the current sensor readings.

4. **User Account -**
   Each user has to create a username and password to use the smart alarm system. User can create the accounts from within the smart alarm application or at mishra14.ddns.net.
   User needs to login to be able to use the application. The application authenticates the user by sending a validation request to the web server and checking the response. The web server uses that request and checks the database to check if the username and password match or not.

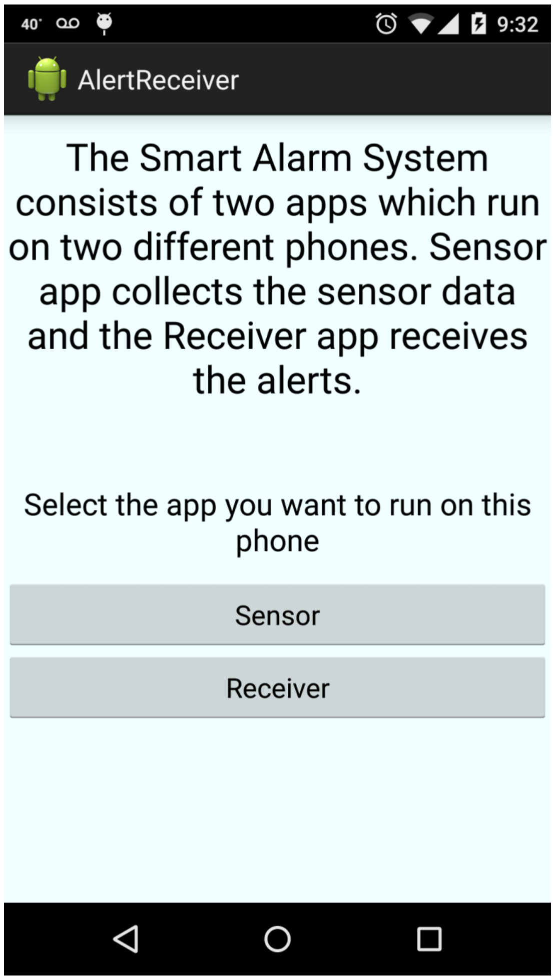




5. **Security -**
   The application is designed with security in mind. The user needs to input username and password for initiating the application. Further, each time the user tries to respond to an alert or toggles the Alert Mode, the user is prompted to enter the password for that username. This password is then used to validate the user before the operation is performed. The validations are done on the server side and all the requests to the server are made via HTTP POST method, for enhanced security.



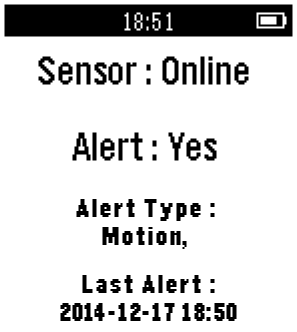
6. **Function Control -**
   The smart alarm system application is designed to function as a sensor and a receiver. The same application has to be installed on both the android phone. The user has a choice to use an the application as a sensor or a receiver dynamically and can be changed at any point of time.



The Smart Alarm System consists of two apps which run on two different phones. Sensor app collects the sensor data and the Receiver app receives the alerts.

Select the app you want to run on this phone

7. **Companion Pebble Application -**
   The smart alarm system comes with a companion pebble application which works independently of the smart alarm system android application and can be used as a monitoring system. The pebble application is designed to fetch the sensor status and recent alert data and allows the user to convenience of monitoring the smart alarm system from their wrist.



Sensor : Online

Alert : Yes

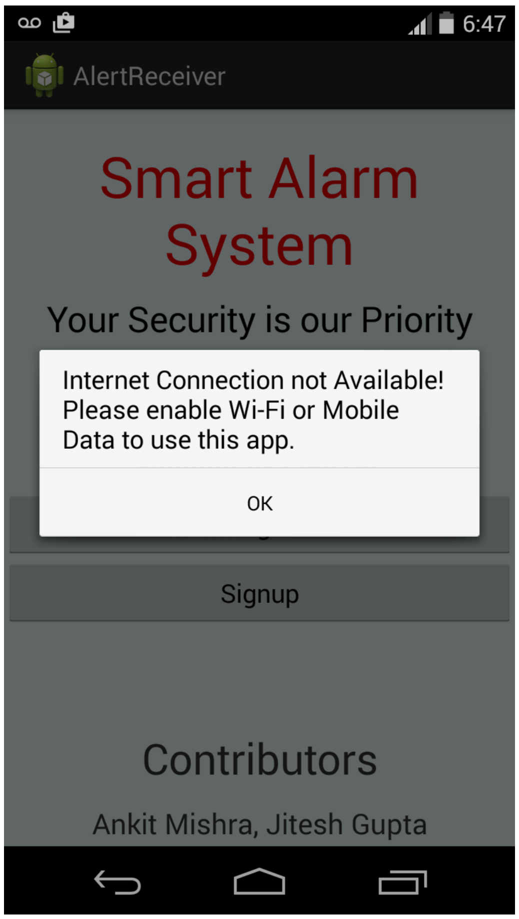
Alert Type :
Motion,

Last Alert :
2014-12-17 18:50

8. Internet Connection Check alerts -

The smart alarm system android and pebble applications are designed to alert the user if the internet onnection to the current device breaks. On the android application, this can happen in two ways. First, if the application is started with no internet connection then the application displays a dialog alert stating the connection problem and then the application quits itself. Second, if the internet connection breaks during the run time of the application then the user is alerted via a notification.

On the pebble application, a message is displayed on the pebble watch if there is no connection to the phone or the internet.


6:47

AlertReceiver

Smart Alarm System

Your Security is our Priority

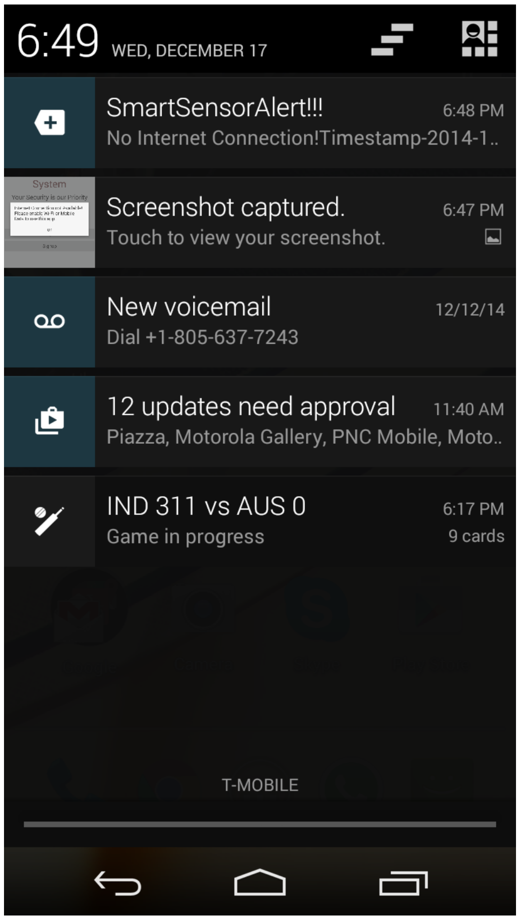
Internet Connection not Available! Please enable Wi-Fi or Mobile Data to use this app.

OK

Signup

Contributors

Ankit Mishra, Jitesh Gupta


6:49 WED, DECEMBER 17

SmartSensorAlert!!! 6:48 PM
No Internet Connection!Timestamp-2014-1..

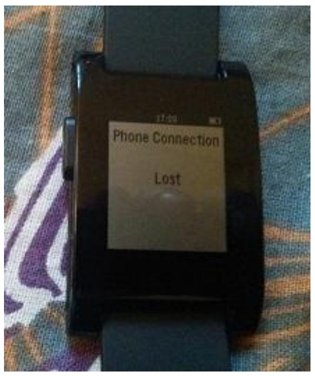
Screenshot captured. 6:47 PM
Touch to view your screenshot.

New voicemail 12/12/14
Dial +1-805-637-7243

12 updates need approval 11:40 AM
Piazza, Motorola Gallery, PNC Mobile, Moto..

IND 311 vs AUS 0 6:17 PM
Game in progress 9 cards

T-MOBILE



CIS 542 - Final Project
Smart Alarm Sensor

Ankit Mishra, Jitesh Gupta

Files Submitted -

1. Project Report - mankit_jiteshg_project/smart_alarm_sensor_Ankit_Jitesh_Report.pdf
2. PHP Source codes - mankit_jiteshg_project/SourceCodes_php
3. Android Source Code - mankit_jiteshg_project/CIS-542-Project
4. Pebble Source Code - mankit_jiteshg_project/SourceCodes_Pebble

SQL Queries -

The following queries were used in the web server to create various tables and updation/insertion of values into those tables.

```
CREATE TABLE user (


    username varchar(30) not null,

    pass varchar(10) not null,

    primary key (username)

);


create table mishra14_sensor (

     ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

     id int auto_increment,

     light float,

     sound float,

     motionX float,

     motionY float,

     motionZ float,

    battery float,

     primary key(id)

);

insert into mishra14_sensor (light, sound, motionX, motionY, motionZ) values (1.0, 60.7, 0.1, -0.2, 0.3);
```

CIS 542 - Final Project
Smart Alarm Sensor
Ankit Mishra, Jitesh Gupta

```
create table mishra14_sense_status
   (
      tsStart TIMESTAMP NULL default NULL,
      tsStop TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
      id int auto_increment,
      light float,
      sound float,
      motionX float,
      motionY float,
      motionZ float,
      status varchar(3),
      primary key(id)
);
insert into mishra14_sense_status (tsStart, light, sound, motionX, motionY, motionZ, status) values (now(), 1.0, 60.7, 0.1,
-0.2, 0.3, 'ON');
update mishra14_sense_status set status='OFF' where status='ON';
create table mishra14_alert
   (
      tsStart TIMESTAMP NULL default NULL,
      tsStop TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
      id int auto_increment,
      idSenseStatus int,
      idSensor int,
      status varchar(3),
     reason varchar(30),
     reasonALL varchar(30),
      primary key(id),
      foreign key(idSenseStatus) references mishra14_sense_status(id),
      foreign key(idSensor) references mishra14_sensor(id)
);
```

CIS 542 - Final Project
Smart Alarm Sensor
Ankit Mishra, Jitesh Gupta

References -

1. getting Rpi server started -

http://www.instructables.com/id/Raspberry-Pi-Web-Server/?ALLSTEPS

or

http://pihw.wordpress.com/guides/direct-network-connection/

2. resolving apache warning -

http://askubuntu.com/questions/256013/could-not-reliably-determine-the-servers-fully-qualified-domain-name

3. Obtaining public IP for raspberry pi server

no-ip.com

http://alexdberg.blogspot.com/2012/11/creating-public-web-server-on-raspberry.html

http://www.instructables.com/id/Raspberry-Pi-as-webserver/

http://superuser.com/questions/669968/ssh-login-using-no-ip-hostname-or-public-ip-denies-my-access

4. installing phpmyadmin -

http://www.raspipress.com/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/

5. Check public ip -

http://askubuntu.com/questions/95910/command-for-determining-my-public-ip

```
curl ipecho.net/plain ; echo
```

6. sending data to php, via HTTP POST, from android -

http://webtutsdepot.com/2011/11/15/android-tutorial-how-to-post-data-from-an-android-app-to-a-website/

7. Java string tokenizing-

http://pages.cs.wisc.edu/~hasti/cs302/examples/Parsing/parseString.html

8. rPI debian image source -

http://www.linuxsystems.it/2012/06/debian-wheezy-raspberry-pi-minimal-image/

9. Alert dialog with text input -

http://www.androidsnippets.com/prompt-user-input-with-an-alertdialog

10. Password type on edit text -

http://stackoverflow.com/questions/2586301/set-inputtype-for-an-edittext