**Product Name: 49erSense**

**Dhiren Gupta | Shwet Shah | Raj Sheth**

**Product Description**

This is an Android app through which you can control appliances like Locks, Garage doors, Lights at your home. The Home screen gives you the current status of all the devices connected and you can control each device through their respective pages/fragments. You can also check your energy consumption levels using this app. This app also includes the weather information according to the user’s location.

**Using the 49erSense App**

Follow the following instruction to use the app.

**Login Page:** As soon as you open the App you’ll get a screen which gives you an option to login or register. If you’re a registered user you can login or else could complete the registration process and then login using those credentials.

**Home Page:** Once logged in, you’ll see the home screen which will you the status of the all the devices which are connected in your house. Click on the action bar on the left of the screen to access the navigation bar which contains access to the different appliances at your home, weather information and energy consumption fragment.

**Security System**: There are 3 states for the security system: Disarmed, Armed(Stay) and Armed(Away). You can select any of the radio button and update the status of your security system.

**Garage Doors:** This page helps to lock/unlock the garage doors. Just select the check box and press open or close button.

**Lights:** This page helps to control all the lights of the house. For each room you have a switch using which you can turn on/off the lights of the room. There is also the seek-bar for you to control the brightness of the room.

**Locks:** You can lock and unlock all the doors in the house using this page. Just check the box in front of the door you want to close or open. Then press the close or open button.

**Motion Sensors**: Activate and deactivate all the sensors in the house using this page.

**Thermostat**: This page lets you control the temperature inside the house. Select the floor whose temperature you want to know and/or change. Switch to the floor, set the desired temperature of the floor. You can switch between the 3 modes: Heat, cold and off.

**Energy consumption**: This page displays the graph corresponding to your energy consumption.

**Live Feed:** You can see the live feed through the webcam which is placed in your home. There are 2 options for live feed: indoor and outdoor.

**Weather:**  This page gives you weather information about the current location of the user.

At any point, you can navigate to the another page using the action bar provided on the left of the screen.

**Debugging**

If anything in the app doesn’t work you can restart the app. If you still face problems try resetting the gateway (Raspberry Pi).

**Setting Up the Project**

~ Setup the Android Project by importing it!

~ Put the htdocs folder into the opt/lampp path!

~ Put the Client.py, led.py and videoStore.sh files in Raspberry Pi!

~ Download Motion video recording library in Raspberry Pi and change its .conf file with the one given in the submission. Starting the service using appropriate command will start the recording.

~ Put the cookieServer.py and Mapping.java files into the server computer!

**Flow of The Project**

**Android Application**

The java classes of the android application calls the PHP files which resides in htdocs folder of XAMPP server!

* The PHP files also check the connection between Server and Raspberry Pi which is helpful in fault tolerance by checking PiStatus.txt file which is edited by cookieServer.py acccroding to the status of the Raspberry Pi.
* PHP files post queries on database according to the values that changed in the android app.
* PHP files after updating database write the same queries on a text file named “newtimestamp.txt”

**Server**

On the server computer XAMPP server is running which has mySQL database with phpmyadmin GUI.

* NMAP is run in order to find the ip address of the pre-registered Raspberry Pi
* Mapping of network is done by Mapping.java in order to dynamically find IP of the Raspberry Pi with pre-registered MAC address of the the
* Server Script named cookieServer.py continuously checks in the newtimestamp.txt file is updated. If the file is updated it opens the socket and sends the data from newtimestamp.txt.

**Raspberry Pi**

Raspberry Pi also has a mySQL database with myphpadmin GUI.

* The Client.py script is run on Raspberry Pi which completes the socket connection on Raspberry Pi’s side.
* It receives the query sent by the server and post it on it’s own local database.
* The Client.py script also post neccessary variables on a text file which a file named LED.py polls, using which it toggles the LEDs according to the data sent by the app as proof of concept!
* Raspberry Pi has USB webcam connected with it which using utility on Raspberry Pi called motion, captures the video. Stores it on Raspberry Pi in .avi format.
* Video from Raspberry Pi to Server is sent using Secure Copy protocol and from the server the video is being streamed on the entire network on a predefined port (8081 by default). If modem’s IP is defined with the port the video can be seen from outside of the server as well!