TEAM SCHFIFTY-FIVE OPENREMOTE TESTING PLATFORM

WHAT IS OPENREMOTE?

The OpenRemote project is a "software integration platform for residential and building automation." Users can create their own apps or console to control and manage autonomous tasks within a building.

OpenRemote also allows developers to build scripts, rules, and events to further automate OpenRemote.

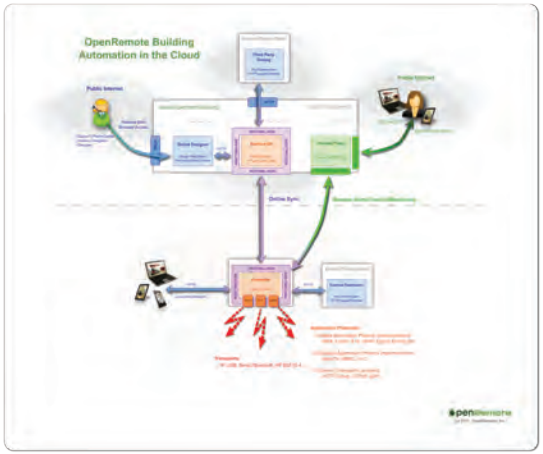
Interfaces are available for iOS, Android, and desktop platforms in order to provide users with a great deal of flexibility when designing automation features.

SUPPORTED PROTOCOLS:

KNX	Infrared	Lutron
TCP/IP	Serial/RS-232	Insteon
Telnet	X10	1-Wire
HTTP/REST	UPNP	

CLOUD BASED DESIGNER

OpenRemote



TESTING FRAMEWORK

The testing framework developed for the OpenRemote, specifically the Controller component, is a lightweight python script that automates all features of the testing process. It is highly robust and can be repurposed by changing only the test cases themselves. A complete test case includes a template file defining the properties of the test, a Java source file that will run code from the project and output a result, and an oracle file containing the expected output of the corresponding test case executable. The script for running the project will form each test case based on the data within each template, build and execute the actual test drivers themselves, compare the output to the oracle, and generate a comprehensive report in order to provide the user with as much relevant information about a test as possible. The testing suite has virtually no runtime memory limits and can handle any number of test cases, allowing developers to define as many as necessary regardless of hardware requirements; the only restriction is the size of the machine's hard drive, as the suite generates miniscule temporary files as part of the script execution.

TESTING REPORT

The output for the complete test case is generated into a well-formed HTML document. Test cases that ran, passed, failed, or with errors are documented within this file. Information about the cases include the source files used for the test cases, the driver of each test case, requirements for the test cases, the component that is being tested, required inputs, the outcome, and a pass or fail for each test. Also, there is a summary that is time elapsed over all the tests and percentage completed, failed, or passed.

Schfifty-Five

TEAM MEMBERS:

Jason Daniel
Chad Hobbs
Brett Ostwalt
Andrew Rodman
Jacob Wisse

CSCI 362 FALL 2012