Design and Implementation of a WiFi Based Home Automation System

Ahmed ElShafee, Karim Alaa Hamed

Abstract—This paper presents a design and prototype implementation of new home automation system that uses WiFi technology as a network infrastructure connecting its parts. The proposed system consists of two main components; the first part is the server (web server), which presents system core that manages, controls, and monitors users' home. Users and system administrator can locally (LAN) or remotely (internet) manage and control system code. Second part is hardware interface module, which provides appropriate interface to sensors and actuator of home automation system. Unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface modules as long as it exists on WiFi network coverage. System supports a wide range of home automation devices like power management components, and security components. The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

Keywords—Home automation, Wireless LAN, WiFi, MicroControllers

I. INTRODUCTION

A. Overview [1]

NOWADAYS home and building automation systems are used more and more. On the one hand, they provide increased comfort especially when employed in a private home. On the other hand, automation systems installed in commercial buildings do not only increase comfort, but also allow centralized control of heating, ventilation, air condition and lighting. Hence, they contribute to an overall cost reduction and also to energy saving which is certainly a main issue today.

Existing, well-established systems are based on wired communication. Examples include BACnet, LonWorks and KNX [1]. Employing a traditional wired automation system does not pose a problem as long as the system is planned before and installed during the physical construction of the building. If, however, already existing buildings should be augmented with automation systems, this requires much effort and mush cost since cabling is necessary.

Obviously, wireless systems [1] can come to help here. In the past few years, wireless technologies reached their breakthrough. Wireless based systems, used every day and everywhere, range from wireless home networks and mobile phones to garage door openers. As of today, little comparative research of wireless automation standards has been done, although such knowledge would provide valuable information to everyone looking for the most suitable system for given requirements.

Ahmed ElShafee, Ph.D., Assistant Professor, Faculty of Computer Science and IT, Ahram Canadian University, 6th October City, Giza, Egypt. & Member IEEE, aelshafee@ieee.org
Karim Alaa Hamed, B.Sc., Teaching assistant, Faculty of Computer Science and IT ,Ahram Canadian University, 6th October City, Giza, Egypt., karimalaahamed@hotmail.com

B. Features and benefits of home automation systems [2]

In recent years, wireless systems like WLAN have become more and more common in home networking. Also in home and building automation systems, the use of wireless technologies gives several advantages that could not be achieved using a wired network only.

1) Reduced installation costs: First and foremost, installation costs are significantly reduced since no cabling is necessary. Wired solutions require cabling, where material as well as the professional laying of cables (e.g. into walls) is expensive.

2) Easy deployment, installation, and coverage: Wireless nodes can be mounted almost anywhere. In adjacent or remote places, where cabling may not be feasible at all, e.g., a garden house or the patio, connection to the home network is accomplished instantly by simply mounting nodes in the area. Hence, wireless technology also helps to enlarge the covered area.

3) System scalability and easy extension: Deploying a wireless network is especially advantageous when, due to new or changed requirements, extension of the network is necessary. In contrast to wired installations, additional nodes do not require additional cabling which makes extension rather trivial. This makes wireless installations a seminal investment.

4) Aesthetical benefits: As mentioned before, placement of wireless nodes is easy. Apart from covering a larger area, this attribute helps to full aesthetical requirements as well. Examples include representative buildings with all-glass architecture and historical buildings where design or conservatory reasons do not allow laying of cables.

5) Integration of mobile devices: With wireless networks, associating mobile devices such as PDAs and Smartphones with the automation system becomes possible everywhere and at any time, as a device's exact physical location is no longer crucial for a connection (as long as the device is in reach of the network).

Typical examples include an engineer who connects to the network, performs a particular management task, and disconnects after having finished the task; or control of blinds using a remote control.

For all these reasons, wireless technology is not only an attractive choice in renovation and refurbishment, but also for new installations.

II. SYSTEM ANALYSIS

A. Problem definition

Home automation systems face four main challenges [3], these are high cost of ownership, inflexibility, poor manageability, and difficulty achieving security. The main objectives of that research is to design and to implement a cheap and open source home automation system that is capable of controlling and automating most of the house appliance through an easy manageable web interface to run

and maintain the home automation system. The proposed system has a great flexibility by using WiFi technology to interconnect its distributed modules to home automation server. That will decrease deployment cost and will increase the ability of upgrading, and system reconfiguration.

System will make use of secure wireless LAN connections between distributed hardware modules and server, and secure communication protocols between users and server.

B. Proposed system feature

The proposed system is a distributed home automation system, consists of server, hardware interface modules. Server controls hardware one interface module, and can be easily configured to handle more hardware interface module. The hardware interface module in turn controls its alarms and actuators. Server is a normal PC, with built in WiFi card, acts as web server. The webserver software is developed using asp.net technology, so web server should support asp application and.net frame work 4.0, like IIS7.0 for windows OS.

System can be accessed from the web browser of any local PC in the same LAN using server IP, or remotely from any PC or mobile handheld device connected to the internet with appropriate web browser supports asp.net technology through server real IP (internet IP).

WiFi technology is selected to be the network infrastructure that connects server and hardware interface modules. WiFi is chosen to improve system security (by using secure WiFi connection), and to increase system mobility and scalability. Even if, user intends to add new hardware interface modules out of the coverage of central access point, repeaters or managed wireless LAN will perfectly solve that problem.

The main functions of the server is to manage, control, and monitor distrusted system components, that enables hardware interface modules to execute their assigned tasks (through actuators), and to report server with triggered events (from sensors).

In setup mode, user can add and remove hardware interface modules, and can create basic macros involving simple triggers and to customize the macros to perform complex series of events. Macros can be activated manually or as a reaction for certain trigger like motion sensors and surveillance cameras. User can also program macros to activate at random; this feature allows your system to turn the lights on and off at random or semi-random intervals.

In running mode, if hardware interface modules report server with received events and execute their pre-programmed macros.

Hardware interface modules are directly connected to sensors and actuator through direct wires connections. Hardware interface modules has the capabilities to control energy management systems like lighting, thermostats and HVAC (heating, ventilation, and cooling) systems, and security systems (door locks, cameras, motion detectors, fire alarms…).

C. System requirements

The following list gives an overview of the most important requirements of the proposed system

1) User friendly interface: User can easily manage system locally or remotely home automation system, through easy web based interface.

2) Security and authentication: Only authorized user can login to the system (locally, or remotely) in order to manage, control, & monitor. If system detects intruders it should immediately alert the system owner and lock login capability for a while.

3) Low cost per node / High node count: Thinking of building automation, hundreds of nodes may be needed to provide automation. However, the market requires competitive performance (compared to wired networks) to be delivered at this low system cost. Additionally, also protocols need to scale to high node count e.g., ensuring message delivery

4) Large area coverage: Another challenge lies in the fact that devices of a building automation system are dispersed over large areas. Since transceivers must not consume so much power, they cannot be built with a transmission range sufficient for sensors to reach associated controllers or actuators directly. Also, they may rely on an infrastructure of access points and a wired backbone network (or particularly sensitive receivers).

5) System Scalability: Scalability is the ability of a system, network, or process, to handle growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. For example, system upgrade/downgrade by adding/removing hardware interface module should be easy and systematic task.

III. SYSTEM DESIGN AND IMPLEMENTATION

A. Proposed Home Automation System layout

As mentioned the proposed home automation system consists of three main modules, the server, the hardware interface module, and the software package. The following figure (1), shows the proposed system layout.

Secure WiFi technology is used by server, and hardware interface module to communicate with each other. User may use the same technology to login to the server web based application. if server is connected to the internet, so remote users can access server web based application through the internet using compatible web browser.
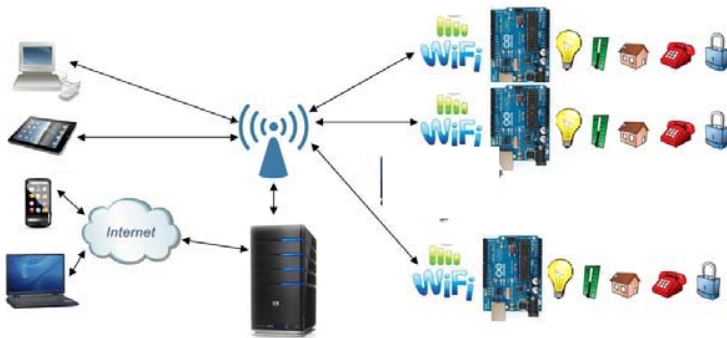
Fig. 1 The proposed home automation system layout

B. Proposed Home Automation System Functions (Home Area Interface)

The proposed home automation system has the capabilities to control the following components in users home and monitor the following alarms;

- Temperature and humidity
- Motion detection
- Fire and smoke detection
- Door status
- Light level
- Video monitoring

The proposed home automation system can control the following appliance;

- Serine
- Lights on/off/dim
- HVAC on/off
- Door lock
- Window shutdown
- On/off different appliance

C. User Classes and Characteristics

The proposed home automation system is designed as a tool for the casual user. A casual user; shall be defined as one possessing general knowledge of the Microsoft Windows operating system and general knowledge of using the Internet by employing a standard browser such as Microsoft Internet Explorer General user; who will have the most use of the system functionality. Administrator; who will control the access and permissions policy of the system, and can add and delete user accounts, anything that a general user can perform, the administrator can also perform.

D. Design and Implementation Constraints

The Proposed home automation system is implemented using ASP, HTML and CSS. The server application is implemented in ASP.Net, and the embedded hardware interface application shall be implemented using C Processing Language.

E. Assumptions and Dependencies

- The component of the system will always be connected
- Each User must have a User ID and password
- There is only one Administrator.
- Server must always run under windows system

- There should be Internet connection available.
- Proper browsers should be installed
- Proper Hardware Components are available
- User is capable of using a computer

F. Software design concept

Software of the proposed home automation system is divided to server application software, and Microcontroller (Arduino) firmware.

The server application software package for the proposed home automation system, is a web based application built using asp.net, Microsoft Visual Studio 2010. Server application software runs on windows OS, requires IIS web server, and ".Net" version 4.0 being installed. The server application software can be accessed from internal network or from internet if the server has real IP on the internet using any internet navigator supports asp.net technology. Server application software is responsible of setup, configuration, maintain the whole home automation system. Server use database to keep log of home automation system components, we choose to use XML files to save system log.

The Arduino software, built using C language, using IDE comes with the microcontroller itself. Arduino software is responsible for collecting events from connected sensors, then apply action to actuators and pre-programed in the server. Another job is to report the and record the history in the server DB.

Figure(2) shows the architecture of the proposed home automation system. The following figure (3) shows classes diagram of proposed system, which consists of five main classes.

G. Classes description

1) dataSourceLog:

Create a log text file in a specific path. This log file has a new log entry, ready to be filled with data. A log entry has the date of the day and a log number.

● writeToLogEntry: Function to write the data into the log file; using a stream writer to convert it and save it into a TEXT file.

● readLog: Read from the file line by line, the log entry with the data

● readNewLog: This function reads the new log entry as just been saved.

2) datasourceXML:

● responsible for all operations regarding the main XML files, User, Sensors and Automation

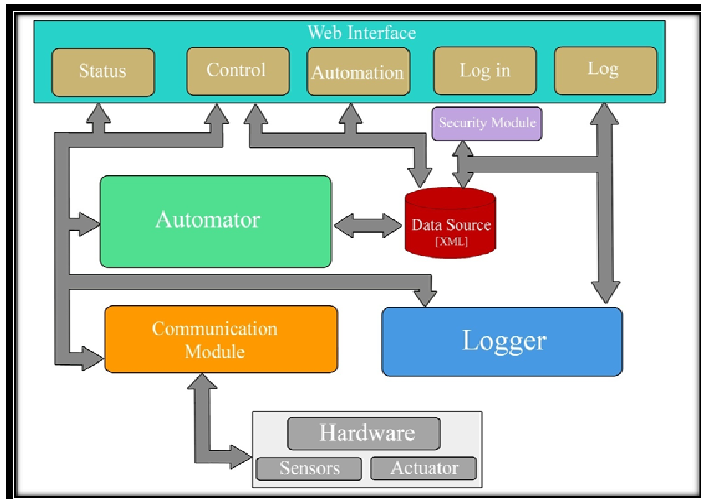● addUser: Function of adding a new user to the system

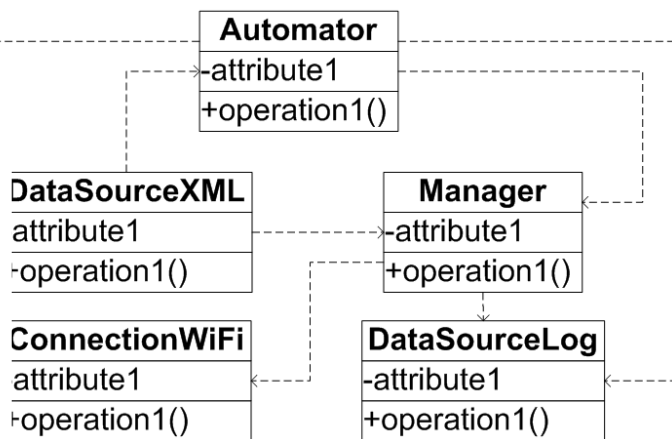Fig. 2 Proposed home automation system architecture

Fig. 3 proposed system class diagram

● getUserHex: User Hex is a security issue. Checking progress will not only be on the username and password, but also on a random generated number for each user. This number changes every time the user logs into the system

● checkUser: Check if the entered username and password is the exact match with the saved XML file or not

● getSensorStatus: Get a sensor last status from the file

● automationTime: Sets the time that the automation is supposed to be activated, in hours, minutes, and seconds.

The following are of the same functionality; which is to change the data of a specific attribute in the XML file given from the function name: automationMotion, automationDoor, automationTemp, automationAppliance, automationDuration, automationSecurity

3) Automator:

Responsible for all automation processes, including the monitoring of the environment for conditions that matches the automation rules configured by the user

● statusInitializer: set all the sensors and actuators back to its initial values. Sensors will be set to 0, and the actuators will be set to the last update in the XML data files.

● getDataFromXML: Get data from XML function

Process: check every one of the sensors and appliance either it contains the key word from the XML file dictionary or not. If it contains the key word it sends the data to check dependency function.

● checkDependency: check if the automation of a specific sensor in an automation is dependent on any other sensor, appliance, or a specific time to be activated.

● watcher: Watcher function acts like a guard; it keeps watching the timer and all other sensors and actuators last updated statuses from the XML file. When all conditions of an automation becomes true. Automation executed.

● processSensor: checks sensors to see the last updates, in case of any dependent automation on one of the sensors.

● activateAppliance: Activates any appliance that should be activated according to the running automation.

4) Manager:

Responsible for maintaining the whole system in general and initializing all that is needed for further operation when the system is first started

● Processor: initializes and load all needed data into the system and establish communication with the hardware module

● addAutomatorObject: load an automation entry into the system to be watched

● deleteAutomation: deletes an automation from the system and XML file and also stops the system from watching it anymore

● getNewAutomation: get all the automation that are assigned to run today

● goManual and goAutomation: toggle the system from manual control to automatic control, needed if the user wishes to stop any automation and have full control

● addAutomatorObject: load an automation entry into the system to be watched

● deleteAutomation: deletes an automation from the system and XML file and also stops the system from watching it anymore

● getNewAutomation: get all the automation that are assigned to run today

● goManual and goAutomation: toggle the system from manual control to automatic control, needed if the user wishes to stop any automation and have full control.

5) connectionWifi:

Responsible for all communication to the hardware module and the data transmission in between

● wifiIntializer: initialize the connection to a specified socket

● writeToSocket: sends data to the specified socket in the form of a string

● readFromSocket: read data from socket after sending it a variable to flag the type of data that is needed to be received

The following are of the same functionality, which is to get the latest status of the specified sensor given in the function name: getDoor, getMotion, getTemperature

The following are of the same functionality, which is to toggle the specified appliance given in the function name to a

new state: switchLights, switchAC, switchAlarm

H.Data Flow

● Login interface → Security Module → Data Source → Logger → Log Interface

To start using the system; the user has to use the login interface to log in into the system. Data passes a security module to be transferred to a 128-bit hex key and checked for availability in the data source represented in XML files. Then the system writes this action into the logger which in turn sends it to the Log interface.

● Status Interface → Communication Module → Hardware Interface → Logger → Log Interface.

Acquiring the status of an actuator or a sensor using the Status interface, is done by receiving these data directly from the communication module, which in turn gets the data required through the Hardware interface represented into the microcontroller. While this process is done and the status is checked repeatedly, any changes will be written in the Logger and appears in the Log interface.

● Control Interface → Communication Module → Hardware Interface → Logger → Log Interface.

Changing the status of an actuator through the Control interface, is done by sending data directly to the Communication Module, which in turn sends it to Hardware interface represented in the microcontroller to apply the required changes. During the process of sending data and changing the actuator, changes are being written into the Logger and appear in the Log interface.

● Control Interface → Data Source.

After changing a status of an actuator and writing it into the log using the Control interface, these changes are being written in the Data Source represented in XML files for later checks.

● Automator → Data Source.

Setting up a new Automation for the system to do is done by the Automation interface, which just saves it into the Data Source XML files; waiting to be activated.

● Status, Control, Data Source → Automator → Communication Module → Hardware Interface → Logger → Log Interface.

The Automator is responsible for executing the automations that have been already saved by the users. In order to do that, it gathers data from the Data Source XML files, Status, and Control interfaces; compares these data with the entered ones in the XML files. Then it sends the appropriate changes required to the Communication Module in order to apply it in the Hardware Interface. During this process changes are being written in the Logger and appear in the Log Interface.

I. Hardware design

The second part of the proposed home automation system design was the choosing of a suitable micro-controller. The requirements for the micro-controller are; a RS232 port, a fair amount of output Digital I/O, and a reasonable speed.
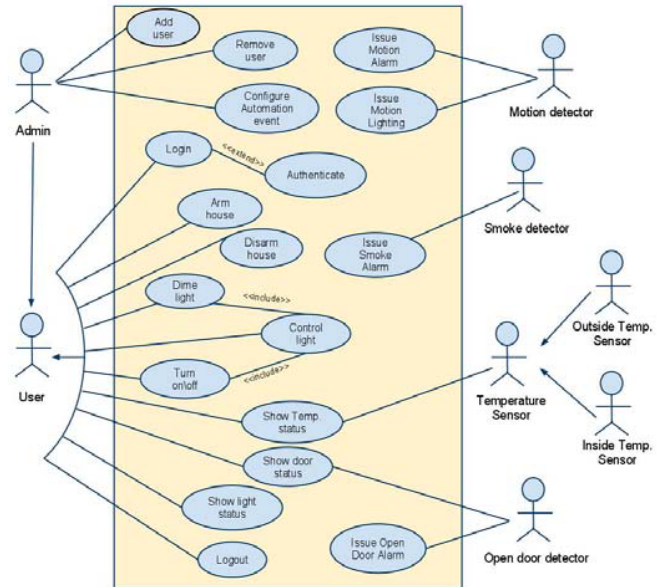

Fig. 4 Proposed system use cases

Also a reasonable amount of EEPROM was needed to enables the system to store device status, and username and passwords related to login, so that data is not lost in the case of power outages.

Arduino is a readymade and open source evaluation Kit based on a 8-bit Atmel microcontroller. Arduino Microcontroller is the core of hardware interface module, which is responsible for exchanging data between home automation server from one side (through WiFi module), and sensors, and actuators from the other side. Arduino communicate with WiFly module through RS232 protocol. If sensors and actuators are directly connected to hardware interface module, an isolating interface is needed to protect Arduino from interference caused by home automation components.

Figure (5) shows the WiFly arduino shield PCB layout, Figure (6) shows the arduino shield PCB layout of a three input alarms. Figure (7) shows the arduino shield PCB layout of a three output actuators.

J. Hardware layout

Hardware consists of four different PCBs, the Arduino PCB (ready-made), WiFi shield PCB, 3 input alarms PCB, and 3 output actuators PCB.
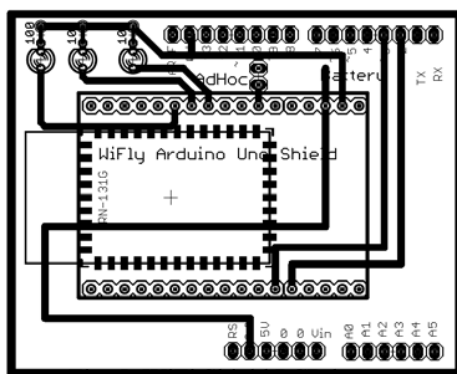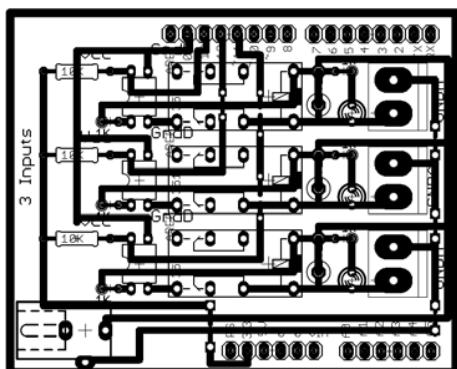
Fig. 5. WiFi Shield PCB layout

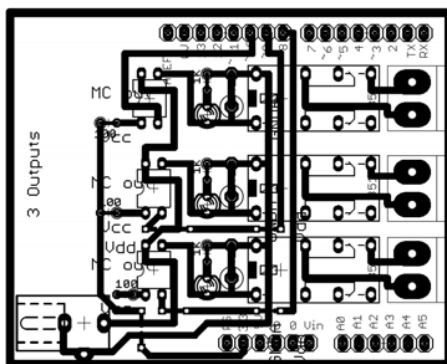Fig. 6 Three input alarms PCB layout

Fig. 7 Three output actuators PCB layout

IV. IMPLEMENTED PROTOTYPE SNAP SHOOTS

Figures (8) and (9) show snaps shots from proposed system user interface. For more information regarding the functionality and features of the proposed Home Automation System, check the prototype published on line in the following link: www.aelshafee.net/HAS01.

V. FUTURE WORK

The following point presents the suggested future work for implemented prototype:

Implements more hardware interface modules, and modify server application software to handle them.

Modify hardware interface module to be able to communicates with sensors and actuators that use wireless technologies like X10, Zigbee, etc.

By doing this system will increase system mobility, configurable, and scalability.

More intelligent should be added to hardware modules to make them capable to take decision according to triggered alarms. Without referring to server for each event and action. That will increase the response time of the system. While hardware interface module reports server with events and actions on pre-programmed intervals.

Replace the WiFly WiFi module with more reliable and stable WiFi module, to increase system reliability.

VI. CONCLUSION

This paper proposes a low cost, secure, ubiquitously accessible, auto-configurable, remotely controlled solution.

The approach discussed in the paper is novel and has achieved the target to control home appliances remotely using the WiFi technology to connects system parts, satisfying user needs and requirements.

WiFi technology capable solution has proved to be controlled remotely, provide home security and is cost-effective as compared to the previously existing systems.

Hence we can conclude that the required goals and objectives of home automation system have been achieved.

The system design and architecture were discussed, and prototype presents the basic level of home appliance control and remote monitoring has been implemented.

Finally, the proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

REFERENCES

[1] Christian Reinisch ,"Wireless Communication in Home and Building Automation", Master thesis, Viennia univeristy of technlogy, Feb 2007.
[2] http://wiki.smarthome.com/index.php?title=Home_Automation
[3] A.J. Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon, "Home Automation in the Wild: Challenges and Opportunities", CHI 2011, May 7–12, 2011, Vancouver, BC, Canada
[4] N. Sriskanthan, F. Tan, A. Karande," Bluetooth based home automation system", Microprocessors and Microsystems journal, issue 26 (2002) pages 281–289, Elsevier Science B.V., 2002
[5] Matthias Gauger,Daniel Minder,Arno Wacker, Andreas Lachenmann,"Prototyping Sensor-Actuator Networks for Home Automation", REALWSN'08, April 1, 2008, Glasgow, United Kingdom.
[6] Malik Sikandar Hayat Khiyal, Aihab Khan, and Erum Shehzadi, "SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security", Issues in Informing Science and Information Technology Volume 6, 2009
[7] D. Greaves, "Control Software for Home Automation, Design Aspects and Position Paper", The AutoHan project at the University of Cambridge Computer Laboratory

[8] Inderpreet Kaur , "Microcontroller Based Home Automation System With Security", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 1, No. 6, December 2010
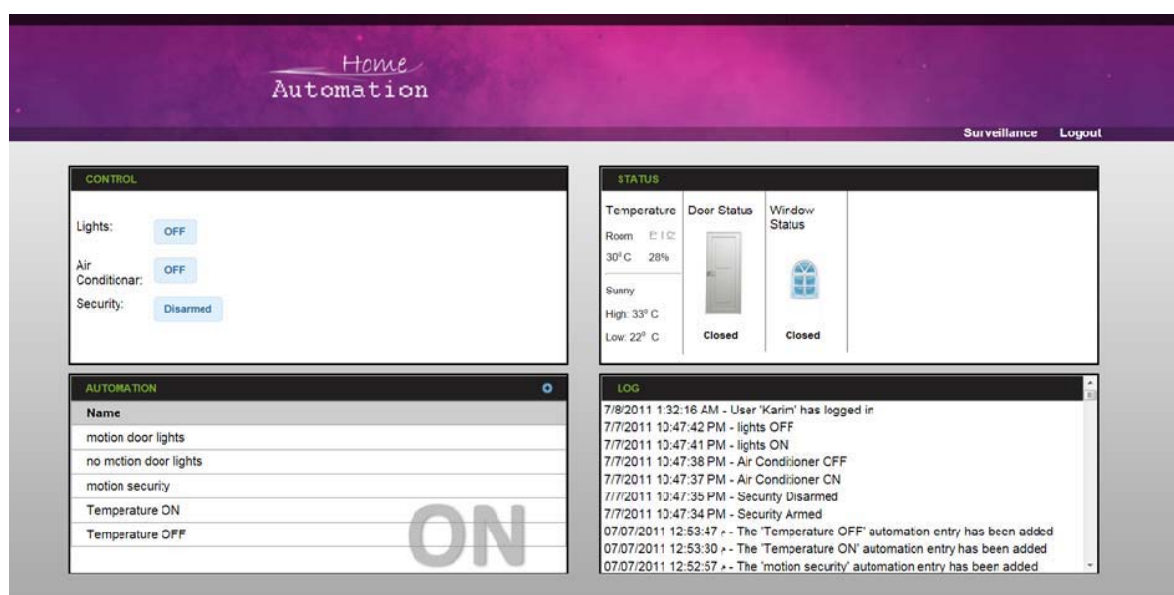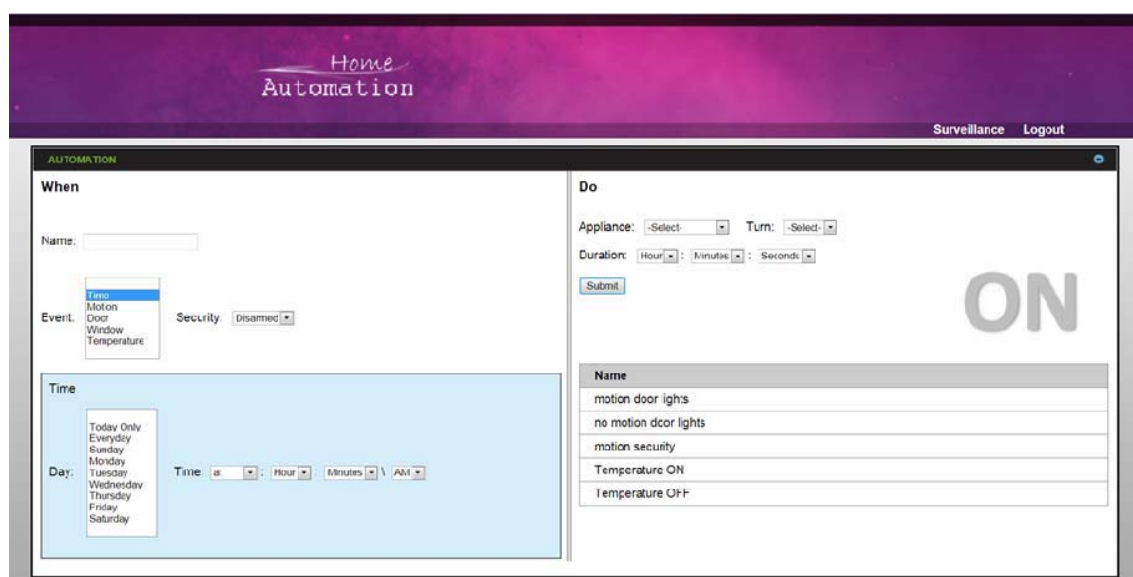
Fig. 8 Overall interface

Fig. 9 Automation interface (system configuration)