

WEB BASED HOME SECURITY SYSTEM

Senior Project

Project Advisor- Prof. Ausif Mahmood

Team Members

**Hector Valentin
Chatura Liyanage
Akshay Pant**

Table of Contents

<u>Topic</u>	<u>Page</u>
1) Abstract	2
2) Introduction	3
3) Research	
3.1 Market Research	4
3.2 Technical Research	5
4) Design Specification and functionalities	
4.1 Specifications	7
4.2 Scenarios	10
5) System Design	
5.1 Design Alternatives	11
5.1.1 Hardware Alternatives	11
5.1.2 Software Alternatives	15
5.2 Design Implementation	17
5.3 Internet Remote Access	18
5.4 Communication between web interface and client PC	21
5.5 Dynamic link library	22
5.6 Database Design	23
5.7 Standalone program	26
5.8 Hardware connecting and wiring	30
6) Testing	
6.1 Code level testing	31
6.2 Modular level testing	32
6.3 System level testing	33
7) Economic analysis, ethical and safety issues	34
8) Conclusion	37
9) References	39

1) Abstract

Imagine a remote system, which would be able to control almost any appliance, ways of entry and exit in your house all from a click in a computer in your house. Now think of being able to do the same via the Internet from any place in the world. This is the innovative Security System.

Innovative Security System deals with primarily three components: a centralized database, a remote application and a web based application. The remote application is customized for clients and gives visual representation of various parts and appliances in the house. The database stores invaluable information regarding the status of various appliances in the house. And the web-based part incorporated the above in a web based application.

The web based GUI makes it possible for users to log in remotely from practically anywhere in the world and be able to control various appliances and parts of a house. The interface will be a very self explanatory interface which anyone with even the most trivial knowledge of the internet will be able to understand and use. This interface has been developed using the HTML, VBScript and ASP based technologies. All data regarding the client's houses is stored in a robust and relational SQL Server database. The hardware part of the house is controlled using a digital board via Visual Basic.

To control various components in the house one could make a "request" using the ASP program or the Visual Basic interface. This request eventually talks to the Visual Basic program which performs the action and sends a record of this change to the ASP program which updates the database.

2) Introduction:

Security is one of the primary concerns when moving into any residential area. Let us consider the issue of security around our town of Bridgeport. Our school is extremely serious about the security issues around the campus. Each student living in the school dormitories is given a PAD (personal alarm device). Upon pressing the PAD the security office is immediately notified about the location and name of the student in distress. This is one factor that completely distinguishes the school campus from any other area in the city of Bridgeport and makes it a safer place for all the school residents. The PAD was one such device that inspired us into developing a security device. Security products are very marketable and if designed, properly developed and efficiently marketed, they can have good economic feasibility in the long run.

Security is not restricted to just residential purposes. This is even more applicable to office workspaces as well. Our project essentially deals with primarily the residential area. The main functionality of our project is going to revolve around areas like controlling (opening and closing) the doors, gate and windows of the house. As an improvement to the project we have added various additional appliances in the house such as a stove, heater, light and receptacle.

3) Research

Our research regarding our product was twofold- Technical research and market research. It was after careful analysis that we came up with our end product.

3.1 Market Research

All people value security. They like to feel safe. We note that people, both at home and at the office, spend a large factor of their budget for security. Most of this budget is spent on security systems. Statistics show that in 2000 every 50 out of 1000 households in America was robbed (www.ojp.usdoj.gov). This was mainly by means of illegal entry by force, such as breaking a window or slashing a screen or by entering through an unlocked door or an open window. Therefore, we have seen a trend in the increasing need for home security systems. Studies have shown that burglary seldom happens at places where an efficient and secure home security system has been setup.

(<http://cognigen.net/protectamerica>)

Through our research we were able to identify common characteristics of a home security system as follows

- 24 hour monitoring
- Ease of use
- Difficulty to hack
- Reliability
- Heat, motion sensors
- Ability to control doors, gates, etc

- System that monitors burglary, fire and medical emergencies
- Efficient, fast and precise notification system

These findings helped us to develop the functionalities we needed to include in our product. Also of these features convenient ease of use was vital, and most people preferred a system that not only can be controlled remotely within the house but from anywhere in the world. Thus the need for a web-based home security system was born.

We have identified two main competitors for our product- namely ADT home security systems and Protect America incorporated. Both of these companies have several products that satisfy most of the features mentioned above. The competitive price ranges were from \$29.99/month- \$799/month. We observed that their market was not only for residences, but also for small and large businesses. Our product is exclusively for residences, and we have offered features such as interactivity and web based control, which our research found is not consisted in our competitor's products.

3.2 Technical Research

We spent a huge amount of time doing technical research to find out the best technology to use in order for us to implement an efficient and scalable solution. Our decisions are discussed in technical aspects in the 'design' section of this document. This section will cover the research we did in order for us to make the correct decisions

- Web browsers- Research has shown that 77% of household PCs used IE5.0 or above and 7% Netscape 4.77 or above on a Windows platform.
http://www.w3schools.com/browsers/browsers_stats.asp) Therefore we designed the product to run on any of these main browsers.
- Operating System- 85% of personal computer users use Microsoft Operating System and related products. (<http://www.w3schools.com/>) Therefore we based our technology on the Microsoft framework. We will use IIS 5.0 as our internet service. It comes with most Microsoft operating system and web programming was done in ASP to run on IIS. Client programming was done in Visual Basic to run on the Windows platform.
- Databases- we did research on many relational databases, such as Oracle, SQL Server 2000, Sybase, Informix and Foxpro. Gartner Dataquest research and benchmarks showed us that SQL Server 2000 and Oracle performance was higher than other relational databases, and had nearly 73% of the market share. We found out that a SQL Server license cost less than a Oracle license and SQL Server integrates better than Oracle into the Microsoft Integrated Development Environment (IDE).
- Hardware- we did research on relays, ISA boards, digital boards, FPGAs and sensor motors to control and monitor the physical elements in the house. We decided to go with digital boards, per Prof. Mahmoods advise, since they were fewer overheads than other PLAs. For example FPGAs are not plug-play devices and, the digital boards we used are, which means as a business we can install our system in a client's house easy and efficiently.

- Internet Service Providers (ISP)- Our research has showed that houses with internet connection is growing from 41% in 1997 to 82% in 2001, and most of those connections are now high speed broadband connections such as DSL, cable, T1, etc (<http://www.commerce.net/research/stats/analysis>)

4) Design Specifications and functionalities

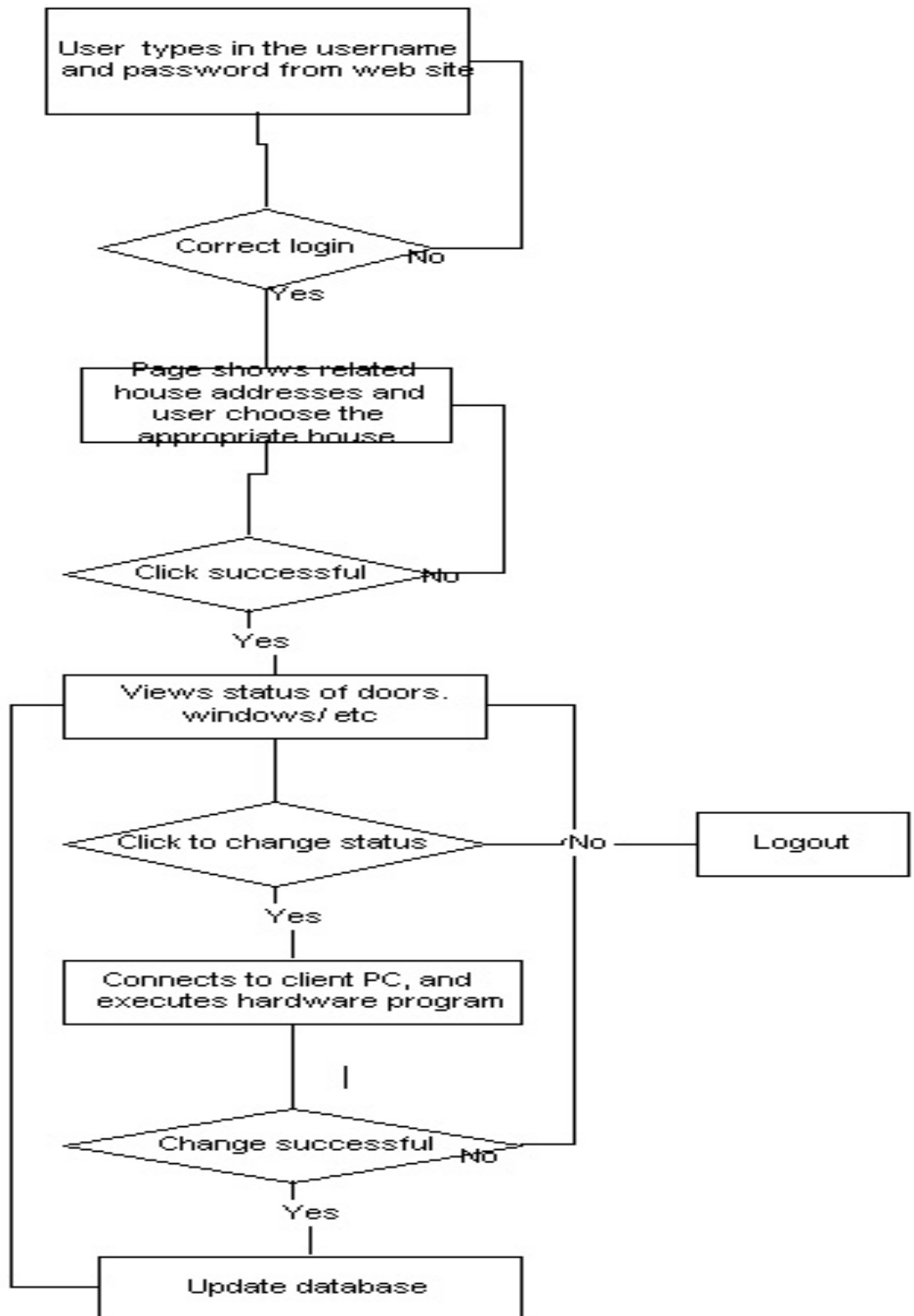
This section discusses the specifications and functionalities that we developed for our product to satisfy. They are written from business standpoint and not a technical standpoint. Our different approaches on how we developed and satisfied these needs are discussed in the ‘design’ section of this document.

4.1 Specifications

- A web based front end where a user can login using a unique username and password.
- After successfully login in the user should be able to monitor status of doors, windows, gates, sensors, lights, etc of each house he/she has registered with our company. The user should also be able to change status of such items such as doors and gates, which in turn will be reflected on the web page
- All connections should be secure and efficient. Servers and databases much be reliable.
- When an illegal change is detected on the client residence the specific change should be recorded in the database, and also phone calls should be made to the

owner, police, 911 and the fire department and also the user should also have the option of being paged, regarding the incident.

- The system should be scalable as to add new products and expand in the future.
- The user should be able to control the system from a program which includes a user friendly and interactive front end installed on client PCs, other than from the web. These changes should also be recorded in the centralized database.
- The web page should be refreshed every 30s so that the user may see the most current information
- A user-friendly front end data entry screen needs to be developed to input data is input into the system, with the registration of new users
- In any security threat, power shut down or server shut down the system should reset itself after resuming services so that the residence is secured and synchronized with the database
- Hardware should be easy to installed and be efficient and non intrusive



Flowchart diagram of user interaction

4.2 Scenarios

Let us now discuss 2 scenarios to make the specifications and functionalities clear

Scenario 1- A user logs in using a unique user name and password to the web site. The user sees his/her house/houses registered with our system, and upon clicking on a house they see the status of each component of that particular house. He/she can change status of some components such as doors and gates. Assume she switch status of a door the client computer would react and the hardware program will change status of that specific door and return the actual result to be updated. The user upon automatic refresh of the page sees the status change.

Scenario 2- a thief breaks in. The program on the client computer recognizes the change and determines it to be illegal. It makes phone calls to the owner, police, 911 and fire department. We also have the option to page the owner. It shuts down all objects, resets the values and updates the centralized database.

5) System Design

Our project consists of implementing a home security system. The system will have the capability to control various house components. For example, the user can open or close electrical doors or can monitor the status of a window. The user can shut-off or power-on electrical appliance. The user will have the ability to perform these tasks either locally from the client's computer or remotely through the Internet.

5.1 Design Alternatives

5.1.1 Hardware Alternatives

In order to send outputs to the house and receive inputs we needed to implement a controller with the capabilities of interfacing with the personal computer's data bus or serial ports. Our first approach was to consider various techniques to implement the controller, we considered using a FPGA, PIC micro-controller and PCI board. The team analyzed the different possibilities to design the controller and consulted our project advisor Prof. Ausif Mahmood and decided to implement the controller using a PCI computer board. The main factor that influenced our decision was the ability to interface the controller with a high level programming language. Also in order to use a FPGA we would have to develop a protocol to communicate between the FPGA and the personal computer serial port. The main advantage of using programmable logic is that they are very high-speed devices. Our application does not require such high speeds. The greatest influence factor in our decision was the cost. We realized that FPGAs and PIC micro-controllers are not plug and play devices. The chips need to be soldered to a board, and electronics components will have to be added in order to reduce noise and attenuation.

This results in a great deal of overhead and delays the time to market for a small start up business. Our final decision was to purchase a digital PCI board. (See figure #1)

www.measurementcomputing.com

PCI-DIO48 Digital Board 48 I/O Bits



Features:

PCI Compatible

- 48 Digital I/O bits,
- Emulates 8255 mode 0
- High output drive

Specifications:

I/O bits: 48

Configuration: 24 I/O bits divided into two 8-bit and two 4-bit ports.

Input specs: TTL family

Output specs: STTL family

Sink: 64 mA min. at 0.55 V max.

Source: 15 mA min at 2.5 V min

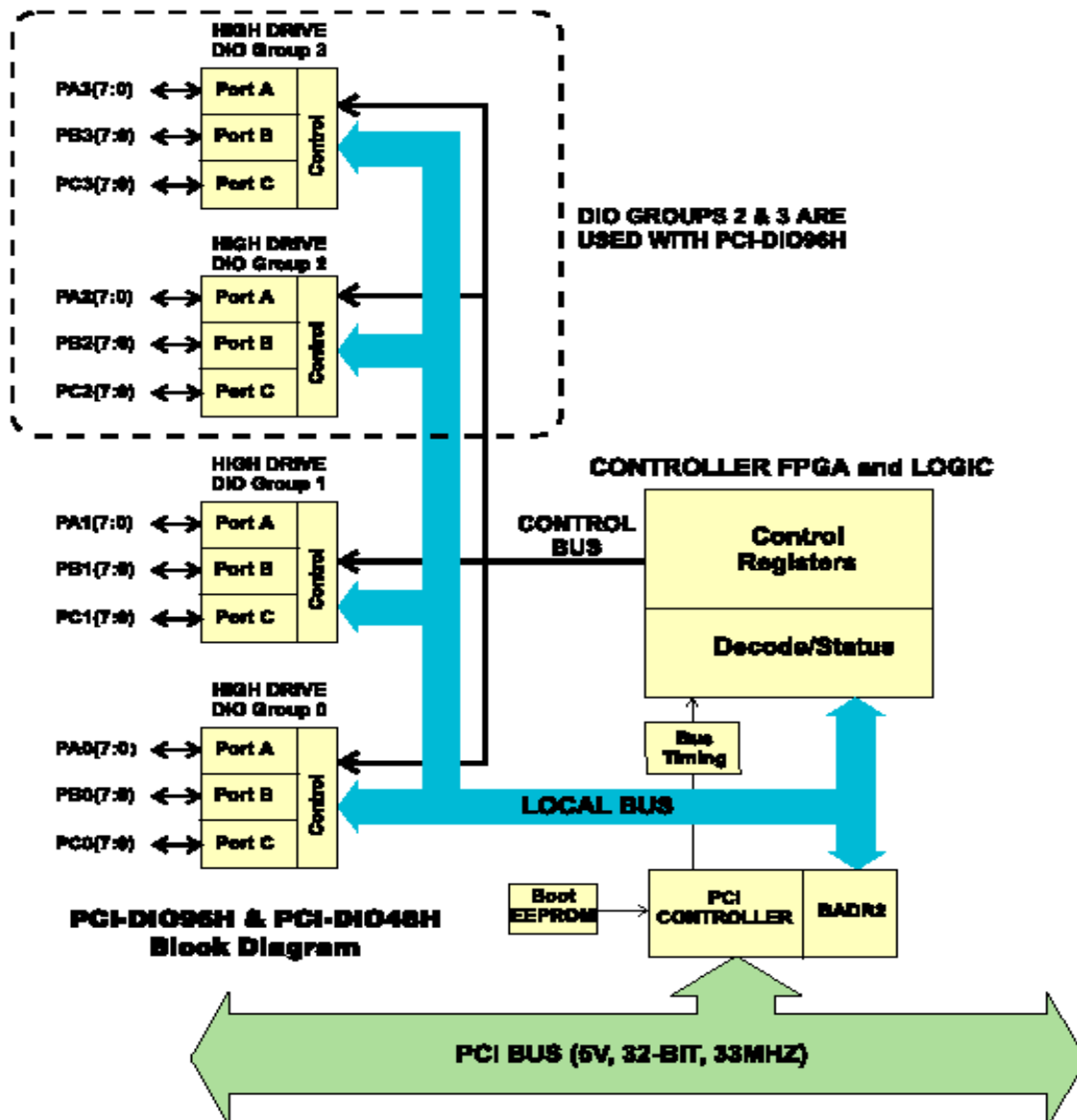
Compatible Accessory Boards:

Solid-state relays: SSR-RACK series

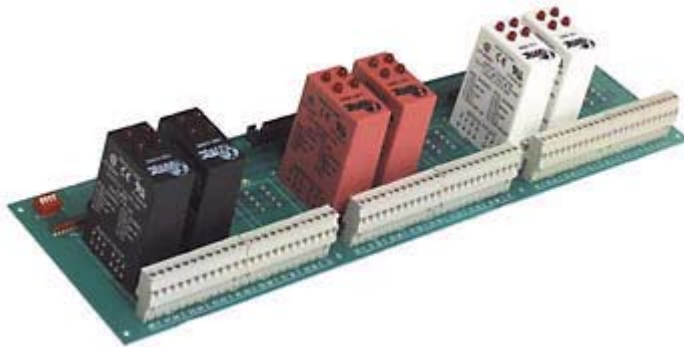
Electromechanical relays: CIO-ERB series

The PCI-DIO48 is a digital board, which consist of 48 digital bits that can be configured as inputs or outputs. The 48 bits are divided into two groups of 24 bits. Each 24 bit group can be subdivided into 4 ports, which are (Port A-8bits), (Port B-8bits), (Port CH-4bit) and (Port CL-4Bits). “The PCI-DIO48 uses 74S244 chips that can sink 64 mA and source 15 mA as output driver .The input buffers are 74LS373 chips and have standard high input impedance of the74LS series devices”. (See figure #2).

(<http://www.measurementcomputing.com/PDFManuals/PCI-DIOxxH.pdf>)



Relay Board



5.1.2 Software Alternatives

Our project has been targeted for the average house owner. The team decided from the start to take the American On Line approach, which is to make the software as simple as possible. The use of clever graphics and icons are very important. Pictures can tell more than words and from the business point of view, small details will catch the interest of the audience.

We utilized Microsoft Visual Basic to develop a very graphical and easy to use interface between the controller and the user. Also since most of the users will be average homeowners and most home computers come from factory with windows operating system, it is very suitable to use Visual Basic on our implementation. Also the PCI-DIO48 digital board has dynamic link libraries (DLL) that make the use of visual basic very convenient.

The Web Base remote access to the security system was implemented using ASP. A relational database was developed using Microsoft SQL Server. The most difficult

decision was how to implement the communication between each individual house and the remote Internet access. The first approach was to design a SQL database and use the database as a mediator between the Internet remote access and the actual house. We realized that the approach was very inefficient for two reasons; first, the house computer will be using its resources at all times even without any information been requested, because a timer will be querying the database all day looking for any changes or requests; second reason is that most hosting servers will not allow clients to hit a database so many times a day. Our second approach was to create an ActiveX component that would run on the Internet browser, and by using sockets, bring an XML page containing the current house status. The previous approach solved the problem of hitting the database very frequently, but the computer resources will be consumed by bringing the status data in a XML page every 20 seconds.

The approach utilized for the project solves both of the problems. The web design will consist of creating a web server on every security system machine using Internet Information Systems (IIS). Every house personal computer will store a file call action.asp. When a request is made the web interface will call the home computer using the IP address and will pass a string of parameters containing the request generated by the web interface. The request will be handled by the file action.asp, which will make use of a Dll control called Control.Dll.house. The DLL file contains all the functions needed to perform all requested actions. Also a database will be use to record all current status on the home security system. Both the web interface and the action.asp will write to the database in order to have updated data on the database.

5.2 Design Implementation

Our design implementation was divided into various modules; web interface, controller interface, database design, DLL development, hardware connection and wiring (see figure #5).

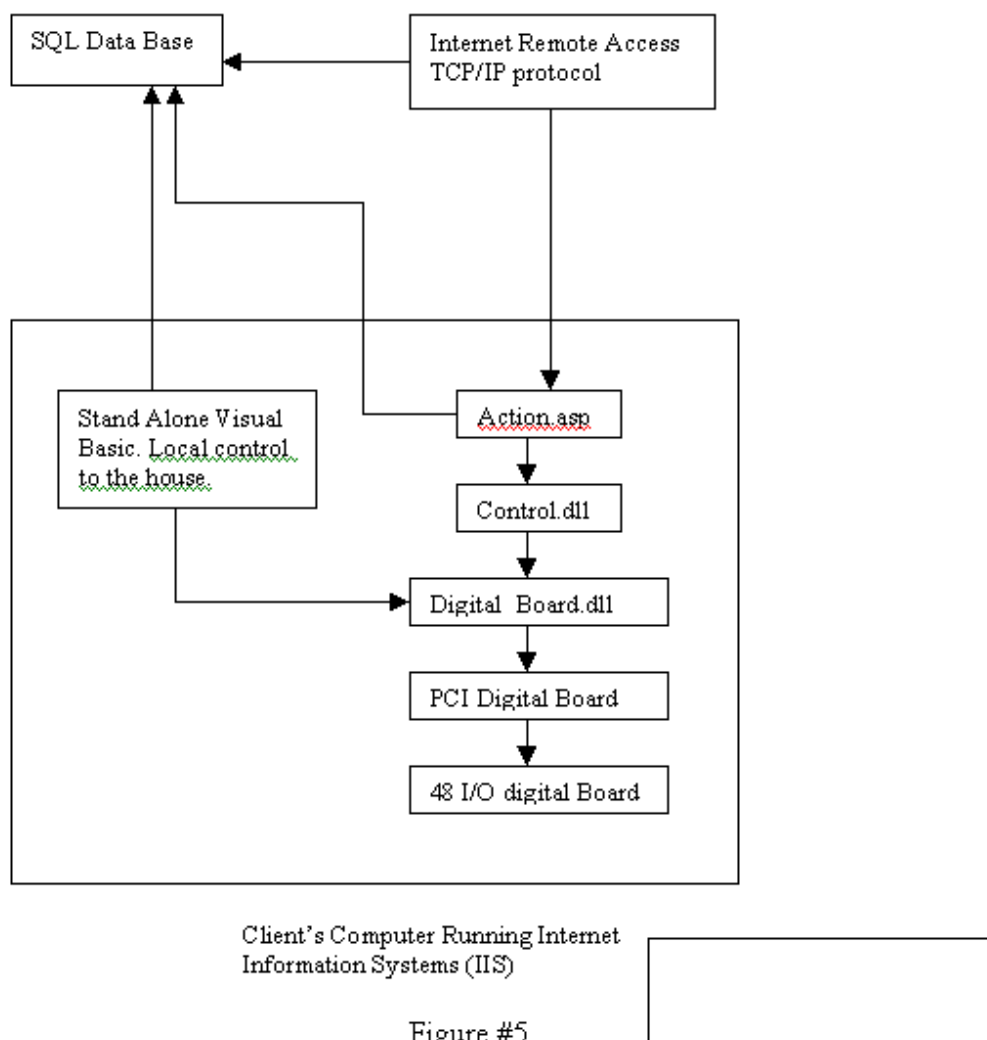


Figure #5

5.3 Internet Remote Access

A graphical web page has been developed making use of ASP and VBScript, which will provide an interface for the user to access, control and monitor the house security system from any location in the world that has an Internet connection. The web interface runs on a server and can be called using http protocol. The web interface consists of three main screens: login screen, destination screen and user input/output screen. The login screen (see figure #6) is provided for security purpose. For our implementation we used a regular http server, which does not provide very strict security. On a real world implementation we would use an https hosting server, which can provide high levels of security. The user logs in the web site by entering name and password. Next, the user will be directed to the select location screen (see figure #7) where the user will be prompted to select the desire location; this process takes care of customers having multiple houses or locations. One the desire location has been selected; the input/output screen will be loaded (see figure #8) on the Internet browser. If the user desires to send any outputs, (for example he needs to open the garage door), a link will be pressed and the web interface which will build a string containing another web address that using the IP address recorded from the select location screen, will call an asp page called action.asp stored on the client's home computer. Besides calling the house computer the web interface will generate a string that will carry the requested operation, which in our case will be door on. For example the web interface will produce the following link:

`http://216.87.103.215/house/action.asp?password=yoyo&switches=2&status=off&type=light&location=front)`

The first parameter is the IP address of the location requested, by having the IP address the web interface will know where is the destination using TCP/IP protocol. The next parameter (action.asp) calls the ASP program that is running on the client's computer. The ASP program contains all the functions needed to handle inputs and outputs. The password is also passed as a parameter in order to verify that the user has the proper security clearance to perform the requested event. The next three parameters determine what type of device is been triggered, in our case is a switch and the switch id is number 2. The status can be either on or off, in our case is calling to turn the light switch off. The type specifies whether the requested item is a stove, door, receptacle or light switch. The last parameter indicates where is the device to be triggered, this field is mostly used to determine whether is a front door or a rear door, it is not used all the time.

Every time a request is made by the web interface, the database gets upgraded with the current status of the house. Also the web interface refreshes every ten seconds and checks the database for any changes. If a status has change it will be displayed on the web interface.

Web Interface login screen

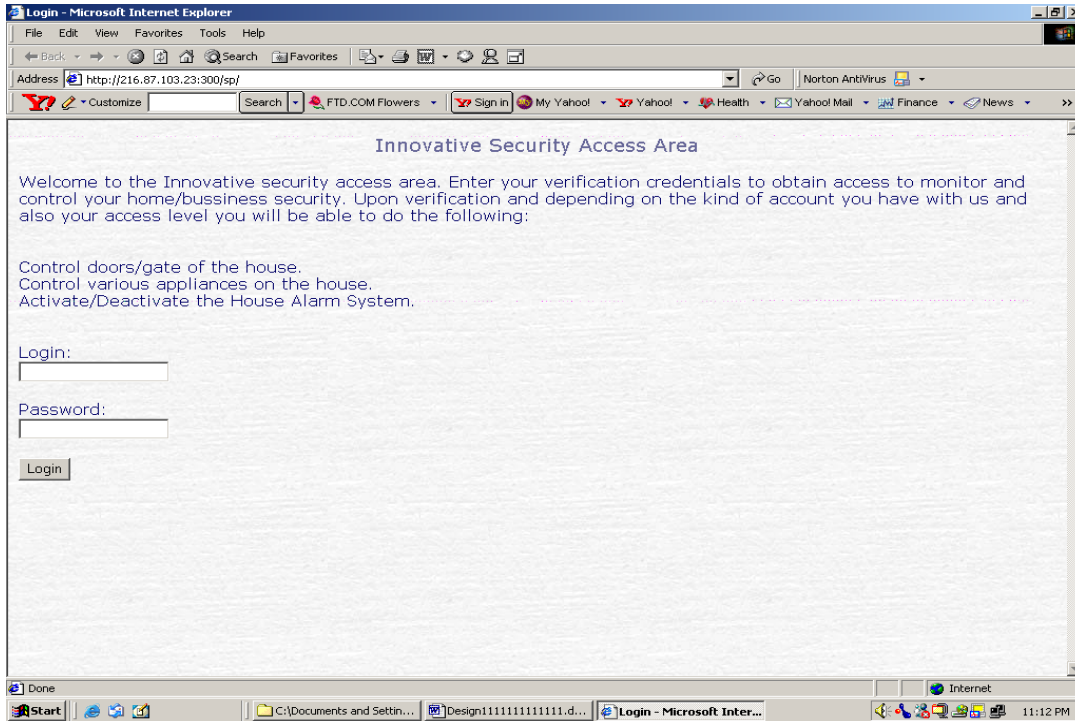
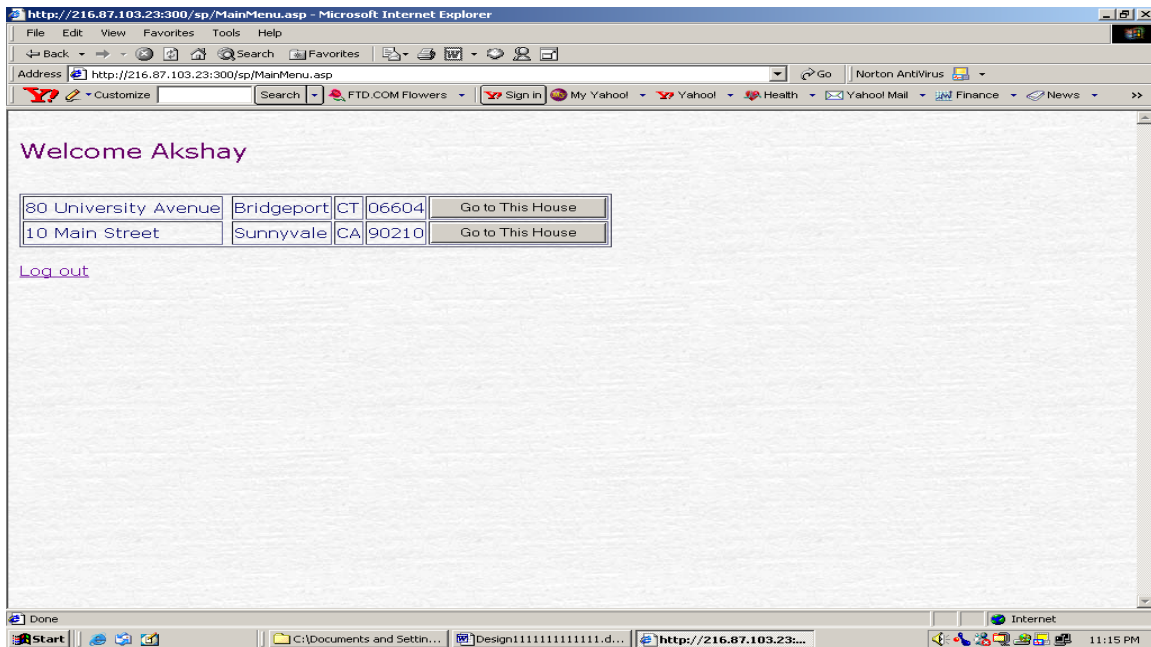


Figure 6

Select Destination screen



Web Interface Input/Output Screen

House to display: 2

Alarm status: **De-activated**

WINDOW	STATUS
kitchen	Off
living room	Off
bedroom	Off
study	Off
room1	Off
room2	Off

Light	STATUS	ACTION
garage	Off	<input type="button" value="Switch"/>
bedroom	Off	<input type="button" value="Switch"/>
bathroom	Off	<input type="button" value="Switch"/>
livingroom	Off	<input type="button" value="Switch"/>
diningroom	On	<input type="button" value="Switch"/>

DOOR	STATUS	ACTION
garage	Off	<input type="button" value="Switch"/>

Toaster	STATUS	ACTION
kitchen	Off	<input type="button" value="Switch"/>

GATE	STATUS	ACTION
front	On	<input type="button" value="Switch"/>

Stove	STATUS	ACTION
kitchen	Off	<input type="button" value="Switch"/>

SENSOR	STATUS
kitchen	Off
bedroom	Off
bedroom	Off
living room	Off

Receptacle	STATUS	ACTION
kitchen1	Off	<input type="button" value="Switch"/>
kitchen2	Off	<input type="button" value="Switch"/>
diningroom	Off	<input type="button" value="Switch"/>
livingroom	Off	<input type="button" value="Switch"/>
bedroom	Off	<input type="button" value="Switch"/>

[Log out](#)

Figure #8

5.4 Communication Between Home Computer and Web Interface

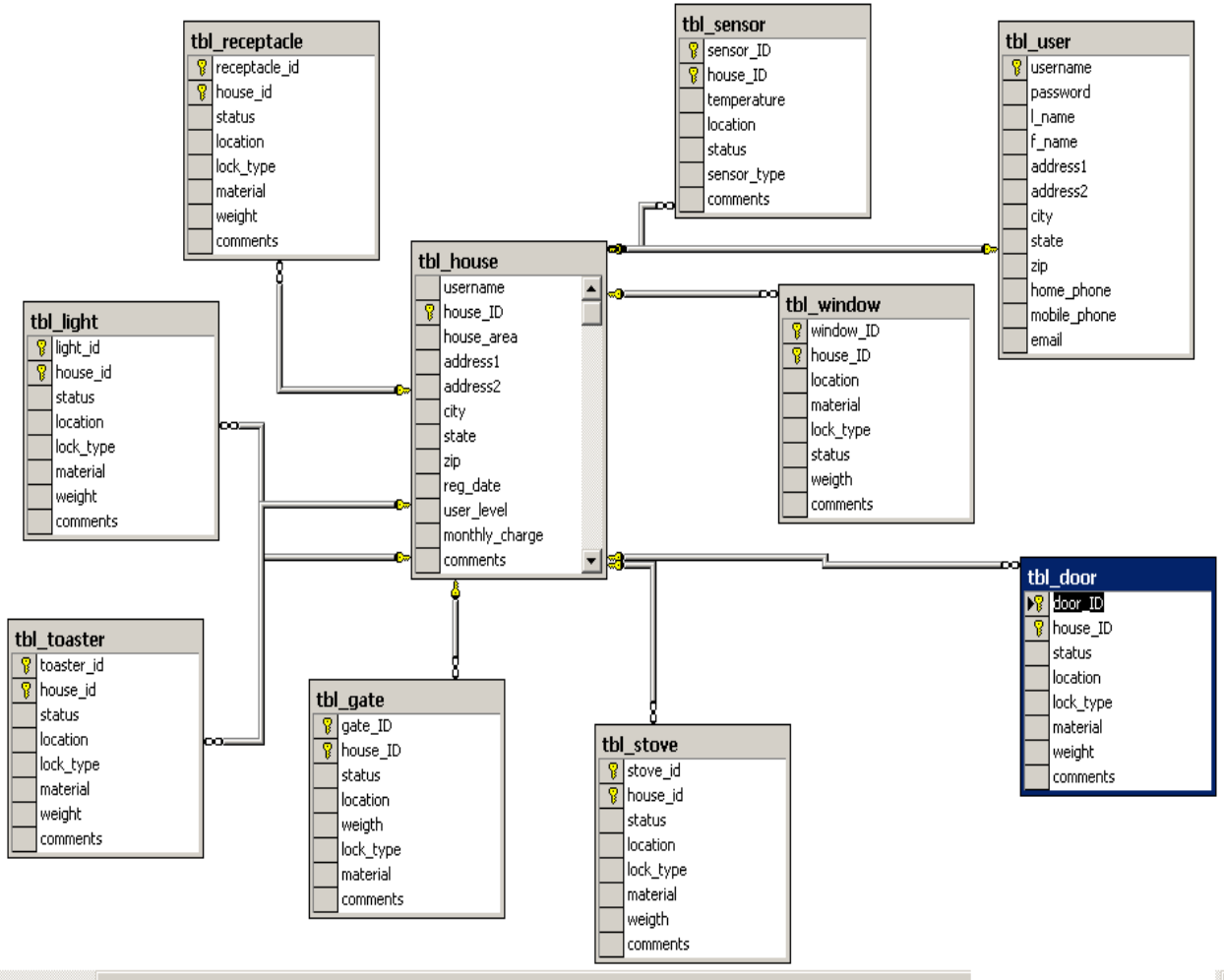
Every client computer will behave as a server using Internet Information Systems (IIS). The client computer will host the file action.asp, which will take care of executing the requests. As discussed earlier the main web server ASP pages will invoke the client action.asp page. Action.asp first established a connection with the DLL and then will take the parameters from the URL string and call the appropriate function in the DLL. For example lets assume stove 2 needs to be switched on. Action.asp will call stoveon() DLL function and send it the necessary parameters. Also the action.asp file writes to the database and updates it. Once the database has been updated the web interface can read the data stored and refresh the page with the latest status. The action.asp file is very

crucial because it makes the link between the web interface and the client computer; it acts as a mediator between the hardware and the web interface by making use of a DLL.

5.6 Dynamic Link Libraries

The Controll.Dll.house provides all the necessary functions needed to communicate with the digital controller board. The DLL contains modules that reference to various Dynamic Library Functions provided by the digital controller board in order to gain access and permission through windows operating systems to directly write to a memory location and to make use of the Digital's board drives. The functions in the controlDll.house is being invoked by action.asp, therefore it waits for a reply back before updating the database with the current status.

5.6 Database Design



The database was implemented using SQL Server 2000. It comprised of 10 tables, and they are discussed below

Tbl_user

The username is the primary key of this table. Every user is registered as a unique user.

This primary key becomes a foreign key in the tbl_house table through a one-to-many relationship. This means one user can have many houses

Tbl_house

The house_ID is the primary key of this table. It holds general information about the house such as address, ipaddress of the client computer, etc. This house_id is connected as a foreign key to the rest of the 'object' tables, making a one- to-many relationship, with each of them. For example one house can have many doors, therefore the houseId will repeated as many times in the tbl_door.

Tbl_door

This table holds data unique to a door. The door_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_gate

This table holds data unique to a gate. The gate_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_receptacle

This table holds data unique to a receptacle. The receptacle_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_light

This table holds data unique to a light. The light_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_toaster

This table holds data unique to a toaster. The toaster_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_stove

This table holds data unique to a stove. The stove_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_sensor

This table holds data unique to a sensor. The sensor_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

Tbl_window

This table holds data unique to a window. The window_ID and the house_ID is the primary key of this table. The house_ID is a foreign key from the tbl_house table. It has a one-to-many relationship with the tbl_house table.

5.7 Stand Alone Program

The stand-alone program is a Visual Basic executable program, which provides full control of the home security system from the actual location. The Visual basic interface runs in the client's computer. The program does not need Internet connection; it has all the functions needed to perform any operation locally. The only devices that the program needs are a working modem and a live telephone line. In order to enter the program the user has to enter a name and password. Once logged in the user can activate or deactivate the system, change options, like who to call in case off an intruder, also can change how much time will the program wait before setting off the alarm when a window or a door has been opened.

The visual basic program uses Dynamic Link libraries to gain access to the drivers on the digital controller board. Also configures the directions of the I/O bits whether they are inputs or outputs.

The Visual basic stand-alone program consists of three main screens. The monitoring screen (See figure #8) presents the user with a top view of the house. The user can monitor the status of every door, window or sensor. The user can also determine whether the alarm is activated or not. If a window is open while the alarm is activated the specific device will turn red, and on the specific green circle will also turn red specifying what is the specific problem. A timer will be initiated and if the user does not enters the password and resets the alarm within a given time, the alarm will sound and a specific phone call will be made automatically to a phone number entered by the user. The program also has the capabilities of sending text messages and emails to a customers email or text pager.

Monitoring program



Figure 8

The output screen (see figure #9) provides control to the house. We believe this is most useful function of the program. It will provide a sense of release for fire hazards and if use correctly can save a significant amount of money in electricity. The Output screen will give you the capabilities of switching the power on or off from local and remote locations. Also the stand-alone program has a history screen where the user can monitor what device triggered the alarm and who was logged in at the moment the incident happened.

Output Screen

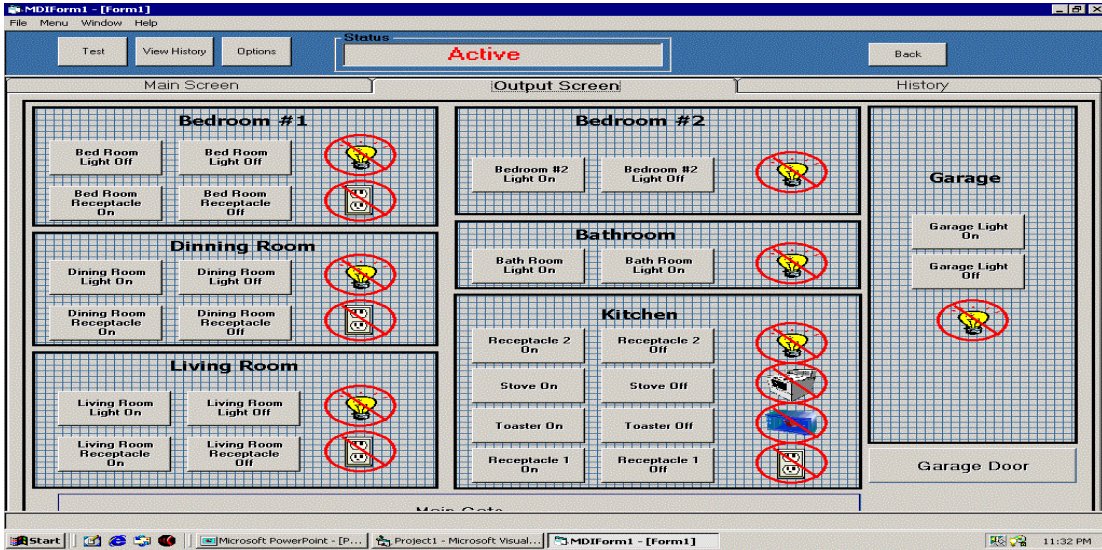


Figure 9

Visual Basic Program Flowchart

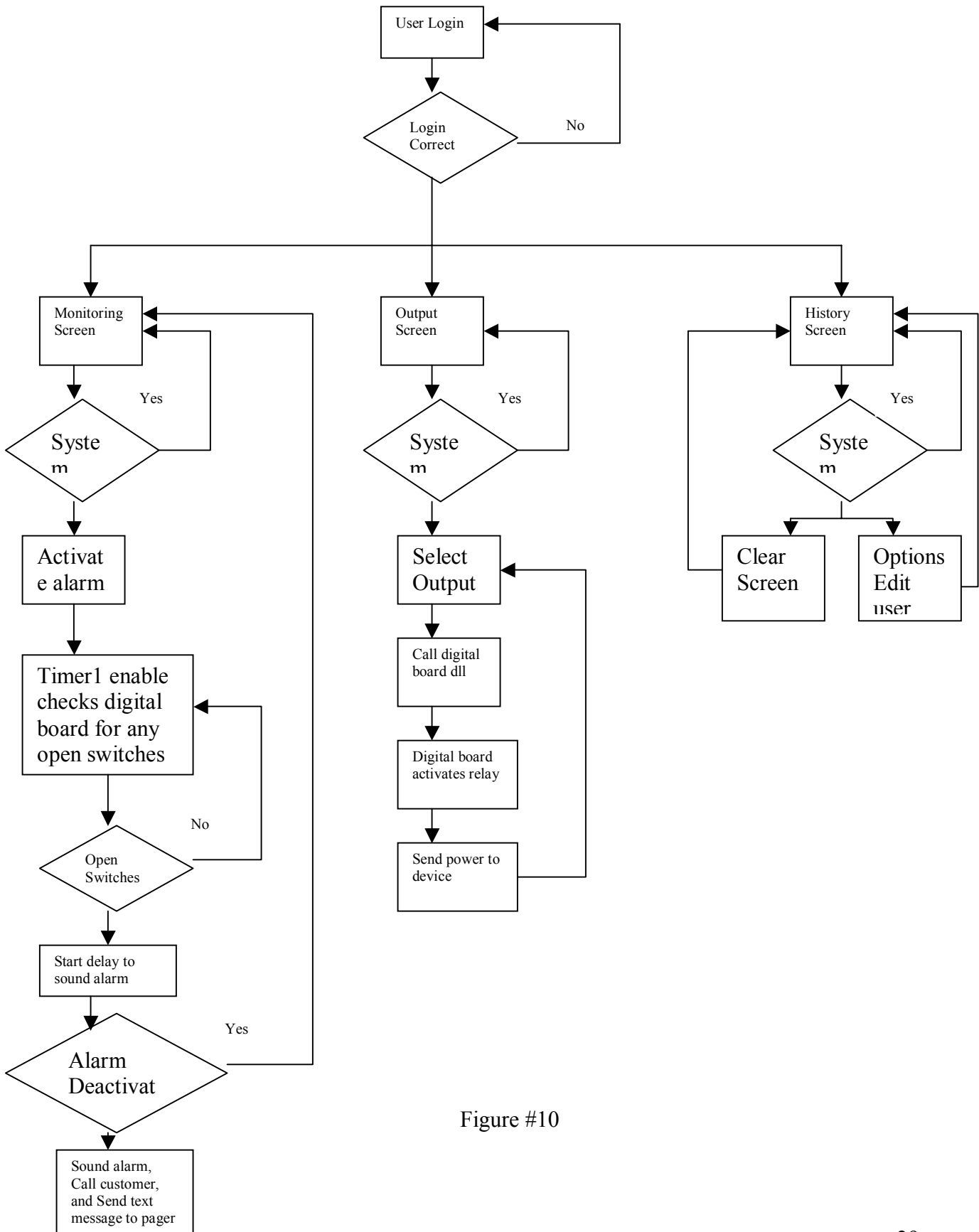


Figure #10

5.8 Hardware Connection and Wiring

In order to show the functionality of our design, we built a small model of a house. We divided the inside of the house into rooms similar to the picture on the monitoring screen. To show the proper operation of the outputs, we connected LEDs and $1k\Omega$ resistors to the quad relays on the relay board. We used a variable quad power supply to feed 12 volts to the relays (see figure #11). When the user clicks on a specific button, that button calls a function that writes a numerical value to a memory location. The digital board uses negative logic. For example a value of 11111010 where the most significant bit starts from left to right means that only bit 1 and 3 must be on and the rest should be off. We used quad input relays to obtain the outputs and supply them to the digital board. We also connected the 12 volts positive supply to the relay's common terminal (see Figure #12). Inserted normally open switches on all the windows and doors. All the input switches are connected to the ground of the power supply. To indicate that a window has been open while the alarm was activated, we close the switch creating a path for the ground signal on the power supply to travel to the relay. Once the ground signal gets to the relay it will create a potential different of 12 volts on the relay, activating the relay, therefore sending a digital signal to the digital controller board creating a positive input.

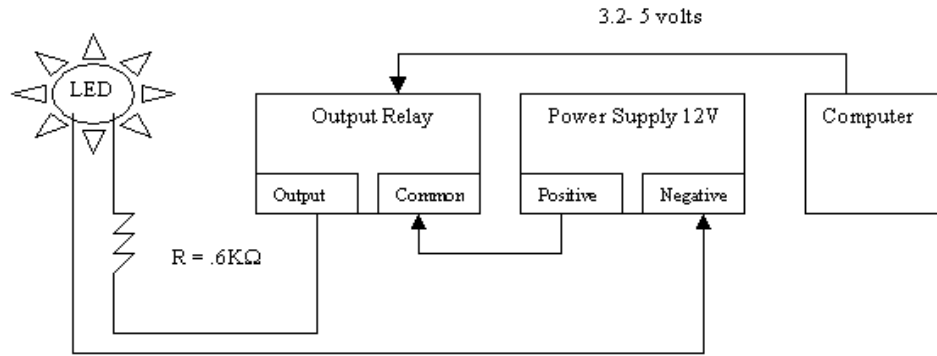


Figure #11

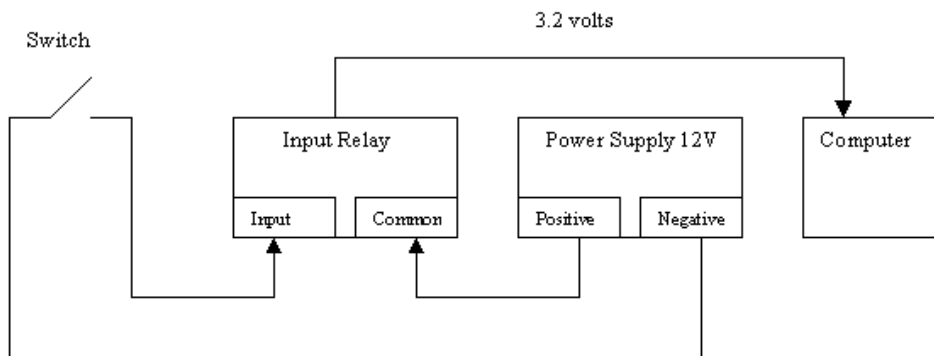


Figure #12

6) Testing

We considered testing to be a very important part of the project and gave very detailed emphasis to it. We carried out thorough test procedures through out the project.

Testing was done in code level, modular level and system level. This section describes how we went about testing on each level and our results.

6.1 Code level testing

Testing was done while developing code. This was done after writing a class or function. We used the Microsoft integrated development environment tools such as Visual Interdev and Visual Studio as code editors. We were able to debug using ‘break point’ techniques and passing values to function prototypes. For example lets say a value was not written to the database. We will put a break point in the function that writes that value to the database, and execute the code stepping from function to function and observing the variable values. We also debugged using the ‘immediate history window’ available with the Microsoft IDE environment. In terms of ASP pages we tested them by simply running them on web browsers. We tested both on Netscape and Internet Explorer. There were several client side coding problems that were associated with these different browsers, such as table alignment and java script errors, which we fixed at this stage.

6.2 Modular Level testing

A module is comprised of one or more functions or in our case ASP pages. We have identified the following as our main modules.

- 1) Web Interface module
- 2) Client’s server’s ASP module
- 3) Database module
- 4) DLL module
- 5) Client GUI interface module

6) 'Sleep' module

7) Hardware Module

We tested these modules using the black box testing methodology. A black box test methodology involved sending inputs into the module and measuring the outputs and comparing it with expected outputs. We do not care what happens inside the module. For example following is a sample query string that the .asp pages will be passing to the action.asp page.

http://216.87.103.164/house/action.asp?password=yoyo&switches=1&status=on&type=door&house_ID=2

so, in order to test the action.asp, we simply called the above query string from a browser, and tested if the output was correct.

In terms of the database module to test the integrity and performance of our database design we generated some dummy data and ran some SQL scripts using Ms query.

6.3 System level testing

After each module was completely tested and debugged we integrated the modules to come up with the complete system. Then we tested the entire system keeping in mind the integration. We also asked different kinds of users who are totally alien to the project to test the system as normal users. Each of tested sections that others wrote so as to isolate any unrecognized bugs. System level integration testing was successful and ran smoothly for most of the system, except for the DLL integration section. Generating, registering

and instantiating a DLL was troublesome and gave many errors when instantiated on action.asp. For example when ever we made a change to the DLL code we had to make the DLL again, however sometime it use to conflict with IIS running, so we had to close down IIS to register the DLL. However we were able to eventually correct all problems. Also we had timing issues at integration testing such as before the door closes, the timer would time out and action.asp page would expect a response back, when it got a negative response, the result was negative though the real case was that the ‘door closing’ procedure had not failed but simply was not finished.

Testing was an on going process for us. It happened in a circular manner. Sometimes, testing will go back to code level testing, when a new functionality is being developed and at any point when a bug is fixed that level is retested from the beginning.

7) Economic Analysis, Ethical Issues and Safety

Ethic’s plays a very important role in the way today’s market is shaped. One while designing a product has to make sure that the product does not infringe on any individuals fundamental rights. This infringement could be on the basis of developing a security system with features similar to a pre-existing product. A wide search on the internet showed us no implication of a product similar to ours. The closest product that we could see was a product called GaurdDvr which basically turns your computer into a video camera such that you can monitor various parts of the house using video cameras. I will mention below about the ethical implications of using a Web Based Security system using the method that GuardDvr uses.

A security company can be very easily sued for invasion of their privacy. While designing the software we had to make sure that in no way would we infringe on ones personal space. This is achieved in our case by not using security cameras as this would lead to the “Big Brother” kind of security system. The primary purpose of a security device is to make sure that everything is safe and secure. Our group firmly believes that the same can be achieved easily without using devices like cameras, microphones and other such devices which might make the residents of a house uncomfortable. This is why we took a path varying from that of GaurdDvr and will be able to patent our product as one different from them.

The goal of our project is to provide the residents of a house with security based on convenience. Let’s describe a simple scenario: A family goes on a vacation to the Bahamas. While in the plane they realize that they have forgotten to close on of the windows in the house. It is not very logical for them to fly back for a thing which is a security concern for them. Using our security system they could use their laptop (or any such computer in local computer cafe in the Bahamas) to connect to the internet and close the window that is making them so uncomfortable.

As seen by the above scenario we can see how our software can help people in matters like this. Some people might argue that we do not really provide security to our customers like the Motion based detector that ADP provides to its customers or security cameras in every room to monitor what exactly is going on. The reason for not having security cameras is that we believe that we can give customers the same sense of security without actually giving them a feeling of being "watched" all the time. And we do not

have motion sensors like ADP is because our product has an alarm based sensor which when activated will alert the correct authorities about intrusion, fire etc.

The market for such a product is big especially considering at how cheap our charges would be. Security if cheap will sell like hot cakes. Who doesn't care about security? No one wants to feel insecure about leaving their belongings vulnerable to external sources (thieves, robbers) or internal sources (fire). Security at a good price is a good investment and will be well worth for a person to buy.

Our security system controls home devices which are analog and need 120 V. However we use digital relays for the same. Initially, we had considered using single relays for control purposes but this proved to be too expensive as each Single relay cost \$20 each. Assuming that we need 48 relays per house (this number depending upon the number of devices the client wants to be monitored) this proved to be a big amount.

The single relay option being too expensive we used quad relays. Quad relays as the name suggests constitute of 4 single relays. However the need of a board to plug these relays into escalated the cost even though it was still less than that of using single relays. Using the same approximation of needing 48 single relays for the house we would end up saving a little over \$200. So, in all the hardware cost along with the cable needed for connection would come to around 800 dollars. This charge would be applicable to client directly as setup/deposit fee + 15% charge as part of our profit.

The first month of service and maintenance would be free. Our electrician would setup the client's computers hardware and relays for a free charge. Following that the client would be paying \$40 a month as a maintenance charge. This charge would in reality be

applicable to our programming cost as development and improvements in the product continue.

Our security system like any other Programming Project is not perfect. It has its own drawbacks and its own security risks. None of these are major or serious enough to impact the project in a big manner. However they do show “holes” in the system which need to be continually worked on till it’s perfected.

References

- 1) (<http://cognigen.net/protectamerica>)
- 2) (http://www.w3schools.com/browsers/browsers_stats.asp)
- 3) (<http://www.w3schools.com/>)
- 4) (<http://www.commerce.net/research/stats/analysis>)
- 5) www.measurementcomputing.com
- 6) www.ojp.usodj.gov/bjs
- 7) www.adt.com
- 8) www.harmonhomes.com
- 9) www.gartner.com/
- 10) Mitchell, Scott “Active Server pages in 21 days”, Sams, Canada, 2000
- 11) Microsoft Corporation whitepapers “Whats new on SQL Server 2000”
- 12) Date, C.J. “Introduction to database Design”, Addison & Weseley, New York,
1998