CHRISTIAN SERVICE UNIVERSITY COLLEGE

HOME SECURITY MONITORING SYSTEM PROJECT
(SAMDAN SECURITIES)

BY

DANIEL KWAKWA-SARPONG
(10140915)

&

SAMUEL SEKYI
(10140807)

PROJECT SUBMITTED TO THE COMPUTER SCIENCE DEPARTMENT OF CHRISTIAN SERVICE UNIVERSITY COLLEGE FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE IN COMPUTER SCIENCE.

STATEMENT OF AUTHENTICITY

We have the greatest, sincere and profound pleasure to declare that we are responsible for any blunder of misquotation and misconception of information found in this study, except any imaginary information that have been deployed inconsiderately from available books and journals.

Declared by

Daniel Kwakwa-Sarpong (10140915) ....

(Student) Signature Date

Samuel Sekyi (10140807) ....

(Student) Signature Date

SUPERVISIOR'S DECLARATION

We hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision laid down by the Computer Science Department of the Christian Service University College.

Certified by

Mr. Christopher A. Abilimi …………………………. ………………………...

(Supervisor) Signature Date

Dr. Thomas Yeboah …………………………. ………………………...

(Head of Department) Signature Date

ABSTRACT

This study is to assist people in need of security measures for their homes to overcome problem that they face in ensuring the safety of their homes and properties. Due to the tense lifestyles of parents today, constant supervision of their children and household can sometimes be demanding. Parents are constantly concerned about the condition of their children and home. They regularly strive to maintain constant child supervision to prevent them from serious injuries. Even though correct safety measures are being taken to prevent serious child injury, one sudden delay of concentration, even for a second, can lead to fatal injuries. A large number of reported accidents with stoves, hot water, swimming pools and so on, have testified that parents need help. The Home Security is a home monitoring device. It is capable of detecting motion in a room, and can inform you of any intrusion or movement within its range. It can also keep you updated on things that go on when you’re out of the house or town.

ACKNOWLEDGEMENT

We would like to express our deepest and profound gratitude to all those who provided us with the possibility and assistance to complete this project. We say thank you to our final year project Supervisor, Mr. Christopher Ayaaba Abilimi, whose contribution and advice helped us to organize this project.

In addition, we would also like to acknowledge with much admiration the essential role of our Head of Department, Dr. Thomas Yeboah for his guidance and direction throughout our entire semester in the university.

We also want to give thanks to our maker and Heavenly Father for the wisdom he has instilled in us throughout our educational life time, we say thank you Lord. Many thanks also goes to our colleagues, the department and lecturers of computer science department of Christian Service University College Kumasi, for their encouragement and support during our years of study.

TABLE OF CONTENT

Content **Page**

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

According to (Johnson, 2014), the home security monitoring system is a sub-system of a security monitoring system whose aim is to help a user monitor his/her household no matter where they are. Apart from having someone home to take care of your house in-case you’re not around, there has to be a way to secure your house and properties. Our home security monitoring system is supposed to keep you updated on whatever is going on in your home anytime and anywhere. With the help of a cellular network it will be very easy to achieve our aims. Cellular home security monitoring systems use a secure GSM signal similar to your cell phone to connect to the home security central monitoring station. There are no wires to cut and it is extremely difficult to break the transmission of the cellular monitoring signal.

Our home is our sanctuary, it is the region of our universe where we feel, or should feel the safest. No matter what is going on in our lives there should always be no place like home. Home security is one of the more complex projects that we want to research, study and work on. We hope by the end of this project we will meet our desires and goals which will educate and broaden our way of thinking as we apply these principles in our daily lives when it comes to home security.

1.2 PURPOSE OF THE STUDY

The main objective of this project is to study the social impacts of security systems in our lives and how we interact with them.

Specific Objectives:

The specific objectives of this project are:

1. To develop an interactive distributive system, which will help ensure security of our home.
2. To provide a cheaper and effective home monitoring system for customers.
3. To evaluate the impact of home security monitoring systems in the country.

1.3 RESEARCH PROBLEM STATEMENT

Many security monitoring systems are being monitored by a monitoring station centre elsewhere by organizations who own them, how can you entrust the safety of your home and properties to people you hardly know. There has been many cases where the people who are supposed to protect and secure your properties are rather harming you instead. Also, there has been instances where you don't get any notifications from these security services until something has already happened. This is because of the lack of an appropriate measure on how we address such issues and the procedures in place.

In response to this problem, our study proposes to investigate several options for making a security monitoring system more reliable. We plan to carry out an all-inclusive participatory investigation into options for providing a less expensive ways to diminish some or all of the problems noted above.

1.4 JUSTIFICATION

The trouble of safety in the country nowadays is very alarming, due to the economic hardship of the country, you’ll only get back from work only to find out that you have been robbed. There cannot be 100% security for your house and properties but there can be away to keep your homes in check all the time.

Some security personnel of today can hardly be trusted to keep your home safe, because there has been a number times where they even aid some of these robbers to rob you, but when it comes to computer software and programs they are assigned to follow a simple instruction and execute without any further thought. This project is specifically designed for the owner of a house or apartment. Imagine that you could check on your house anytime, anywhere and any day to see what’s going on. Justifying the development and implementation of a security monitoring system can be difficult since IT department software purchases are faced with more scrutiny than ever in today's extremely competitive marketplace. We want to develop a very less expensive but effective and efficient home security monitoring system which will make you feel comfortable wherever you are with less worries of your home and properties.

1.5 LIMITATIONS AND DELIMITATIONS OF THE STUDY

1.5.1 LIMITATIONS OF THE STUDY

We went out to find out the challenges that most security monitoring services in Ghana face on the services they provide, but we had a little response from these entities due to the anticipated limitation which included; questions not being properly answered by most security services and the problem they sometimes face, its either they didn’t have enough time due to the nature of their work or they were reluctant to give information due to “oath of secrecy” for some personnel required clearance from their Board of Directors. The fear of disclosing information due to the protection of “corporate secrecy”. Gathering information and data on the performance of these institutions would appear to be an invasion into their corporate secrecy. Despite these issues the study will ensue to treat in the firmest assurance the data and information collected as an academic exercise and also due to limited time constraint and availability of funds the study actually selected few places in the security organizations in the country.

1.5.2 DELIMITATIONS OF THE STUDY

We had an interview with some residents who are already using a home security monitoring system in their homes, and they gave us information based on what we wanted. We observed that most of them don't even know how to operate their security monitoring system well and some highly check on them at all.

But our project only provide a system that can be able to record and give you a notification alarm beep to inform you of an intruder. Our study took as a lot of time to research and understand how most home security systems work and the constraints involved.

Our project does not have a voice input to record audio sounds of intruders. Since we designed it to record only videos of the movement of anybody that comes in contact within the monitoring range. Another delimitation of our project is that it cannot be used to take snap shots or pictures of any motion detected, live video recording is our only specialty.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

Literature review is needed before any project is begun. The review will help to understanding the scope of the project and also the need to build the project. The review comes from the reading on the websites and also from the books. The information from the review will be used to start the project with an excellent idea. The review is also comes from the sample of the existing project in the websites. There are numerous projects that are associated with this project. To make things easier, this section will focus on few projects that are heavily associated with this project.

2.2 PREVIOUS RELEVANT STUDIES AND RESEARCH WORK.

There are many existing devices in market such as CCTV Cameras, IP camera, Infrared Sensor, Laser Sensor and so on.

CCTV.

Implementation of CCTV cameras are very costly and has drawbacks since it require constant monitoring of every activity which is not as ease. Continuous manual visualization hampers the productivity and time. Criminals can penetrate into the CCTV system, thereby facilitating criminal acts.

IP CAMERA.

Implementation of IP cameras are also very costly and not feasible. This system cause major problems as it becomes open to hackers via internet.

INFRARED SENSOR AND LASER SENSOR.

These devices are quite economic in comparison to above devices however they have some drawbacks too. These devices are difficult to install and rarely available. One of the major disadvantages of infrared sensors is the size required to provide good resolution to the signal.

IR MOTION DETECTOR.

A motion detector is a type of electronic security device that senses movement and usually triggers an alarm. Many types of motion detectors can sense motion in total darkness, without an intruder becoming aware that an alarm has been triggered. A motion sensor or movement detector is a device that senses movement to help raise the alarm if an intruder breaks into your home. This ability means that a motion sensor can play a key part in your home's security system.

There are a multitude of sensors that become useful in industry settings, especially where factory operations are concerned. Different sensors include temperature sensors, optical sensors, pressure sensors, motion sensors, ultrasonic sensors, and more. Sensors allow for smooth operations and accurate measurement.

Detector is an electronic circuit employing a transducer that sends an electronic indication of an event or condition to a security system control panel based on sensory input to its detection device from the premise.

Ultrasonic motion detectors have certain advantages and disadvantages when compared with other types of motion detectors. The main advantages are that they are very sensitive and extremely fast acting. However, the largest problem with this type of motion detector is that it sometimes responds to normal environmental vibration that can be caused by a passing car or a plane overhead. Some types of motion detectors use infrared sensors to avoid this problem, but even these detectors have some problems.

2.3 THEORETICAL FRAMEWORK

Software engineering has emerged as a distinct field within computer science, encompassing all the phases of software development. In fact, by now it is an autonomous topic into the wider computer science field (Sommerville I., 2004), divided into numerous distinct sub-topics that all converge to one target: effective, productive, usable, useful, maintainable, dependable and within-a-budget software. Concerning the importance of software nowadays, software engineering occupies the niche of the art of designing and constructing the ever-changing operations and functions of the nervous system of contemporary society. This is not to say that hardware is not important (as if, for example, the human body were considered unimportant for the human society).

However, IT hardware alone would be meaningless, in the exact same way that an aggregation of bodies does not make a human community (Sommerville I., 2011: 264). One cannot overemphasize the importance of software: for it is software by virtue of functionality and potentiality that enfolds the hardware with meaning and brings forth the concept of the networked society, rather than a - socially speaking - indifferent network of machinery (Latham R., Sassen S., 2005); not only software turns the hardware into something usable, but more importantly it enfolds the hardware with contingency, thus entailing a horizon of available selections (Luhmann N., 1995). Those selections of course, refer to the social systems that utilize the software, enfolding the ICT networks with meaning. As Ian Sommerville (2011: i) notes "Software engineering is a critically important technology for the future of mankind". Maybe it would be more accurate to argue that software engineering is critical for the future of the society, the way society itself seems to design it.

This is a crucial point for it is evident that society nowadays considers ICT technologies as a given tool, a "ready-at-hand" (Heidegger M., 2006), and therefore certain social functions realized by the modern computer networks are forming a conceptual infrastructure upon which novel social expectations are emerging. This simply means that the social future is conceptually (and eventually substantially as well)

constructed within realms realized by ICT. The critical role of software enhances the importance of software engineering, bringing it to the centre of the economic, educational and consequently social evolution. Curiously though, software engineering is still based on empirical, trial-and-error procedures, as if it were a totally new discipline that lacks a theoretical apparatus. We argue that this is not the case: software engineering rests upon the fruitful and complex apparatus of cybernetics and systems theory. Disregarding these theoretical underpinnings, results in a multitude of contradictory approaches, unnecessary expenditure of time and resources and quite often - ICT project failures. In this paper, we try to reveal the correlation of second order cybernetics (Forester von H., 2003) with software and systems engineering and to elaborate on the increasing importance of the development of a precise theoretical framework for that discipline.

CHAPTER THREE

METHODOLOGY

3.1 INTRODUCTION

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system. Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, and extreme programming. A methodology can also include aspects of the development environment (i.e. IDEs), model-based development, computer aided software development, and the utilization of particular frameworks (i.e. programming libraries or other tools).

Methodology is the systematic, theoretical analysis of the methods applied to a field of study, or the theoretical analysis of the body of methods and principles associated with a branch of knowledge. It, typically, encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques. A methodology does not set out to provide solutions but offers the theoretical groundwork for understanding which method, set of methods or so called “best practices” can be applied to a specific case.

To develop our Home Security Monitoring System, we will be using the Waterfall Model of the System Development Life Cycle (SDLC). The waterfall model is probably the oldest and the best-known model as far as software development process models are concerned. The waterfall model is so named because it employs a 'top down' approach regardless of whether it is a waterfall model in testing, or a waterfall model in SDLC.

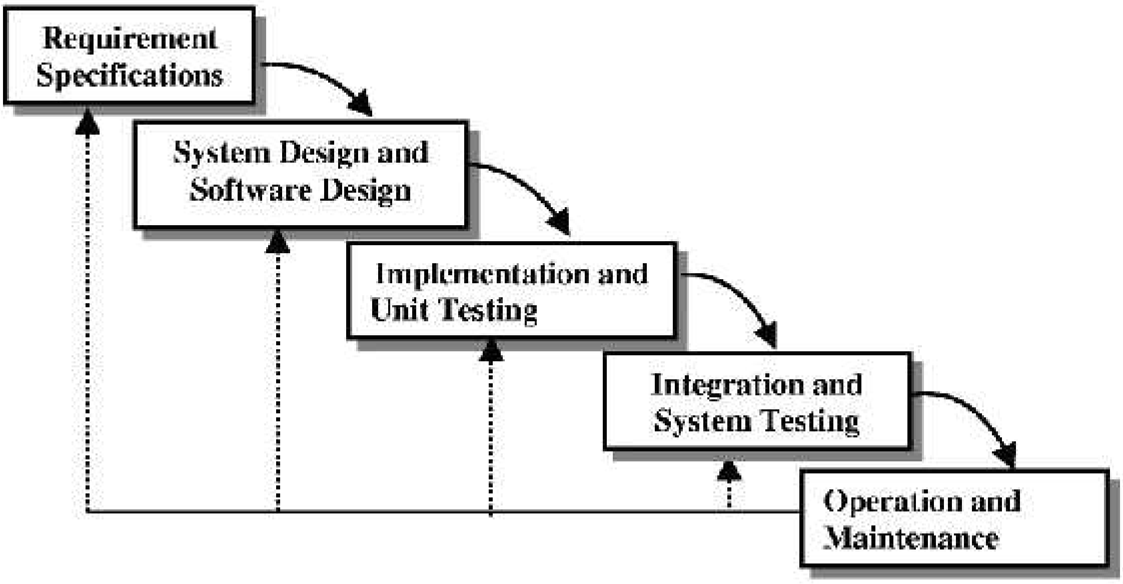

Requirement Specifications
System Design and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 1.1: Waterfall Model

3.2 REQUIREMENT SPECIFICATION

First and foremost, we need to completely analyze the problem definition and all the various project requirements. This phase is commonly referred to as 'Requirement Analysis'. Our Requirement Analysis include the materials we are going to use in developing the system. We have gathered all this information and listed them below.

1. Microsoft Visual Studio C#; This is what we are going to use to build and compile our home security monitoring software.
2. AFroge.NET; This framework is comprised by the set of libraries and sample applications, which we basically need in order to get our video capturing device working perfectly with our software.
3. Motion Detection; A motion detector is a type of electronic security device that senses movement and usually triggers an alarm. A motion sensor or movement detector is a

device that senses movement to help raise the alarm if an intruder breaks into your home. This ability means that a motion sensor can play a key part in your home's security system. Now this is also included in the software to help us achieve our aim.

4. Webcam: A hardware device that is will be used as a security camera for our system and its cheaper than any other security camera.

5. Computer; The computer is going to act as a server to the monitoring of all the activities and processes the system is going to embark on.

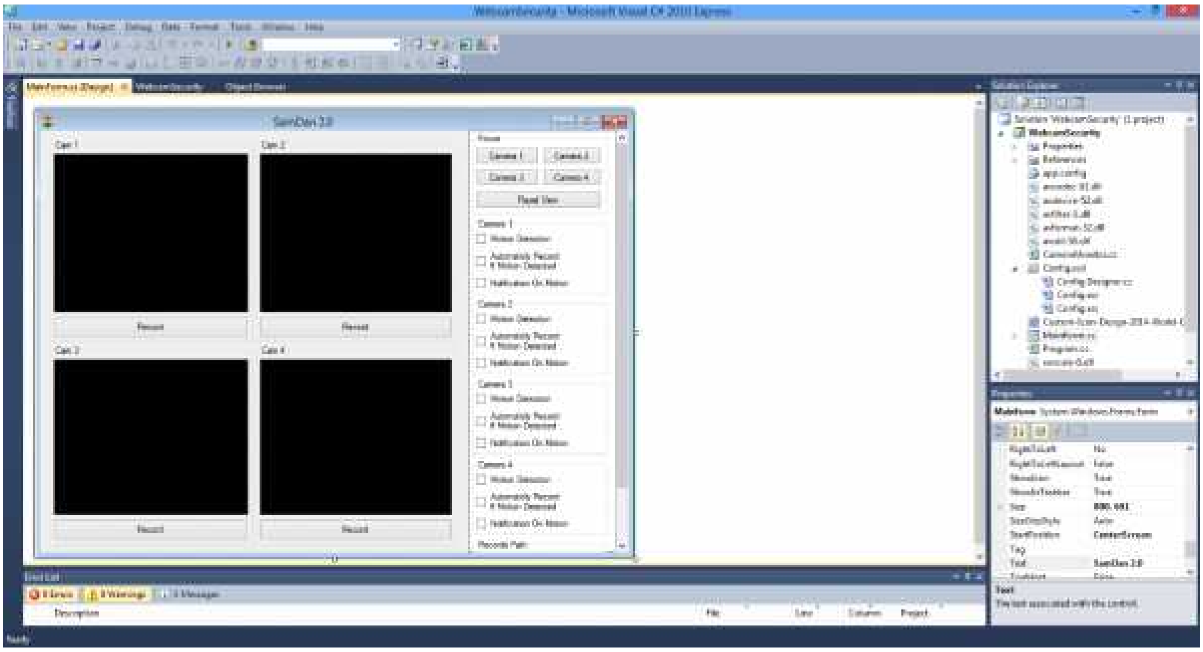
Now we have thoroughly and exhaustively identified and understood all the project requirements, they are to be properly documented, after which you move onto the next phase, which is known as 'System Design'.

3.3 SYSTEM DESIGN

System Design involves specifying and designing the project's hardware and software requirements, and their inter-relation. The entire software aspect of the project is broken down into different logical modules or blocks which are identified and systematically documented.

Our software development construction methods are very simple. We just had to make some references to the AForge.net libraries in order to import them in the Microsoft Visual Studio C# before we were able to start our project development. The name of our home security monitoring system is "SAMDAN SECURITIES" based on the project members names Samuel and Daniel.

Firstly we created our form which will be used as the user interface of the system, we then added some buttons and check boxes which will all act with a specific command when selected. We created up to four cam views of the system, which in other words can support up to four(4) webcams.



RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	880, 681
SizeGripStyle	Auto
StartPosition	CenterScreen
Tag	
Text	SamDan 2.0

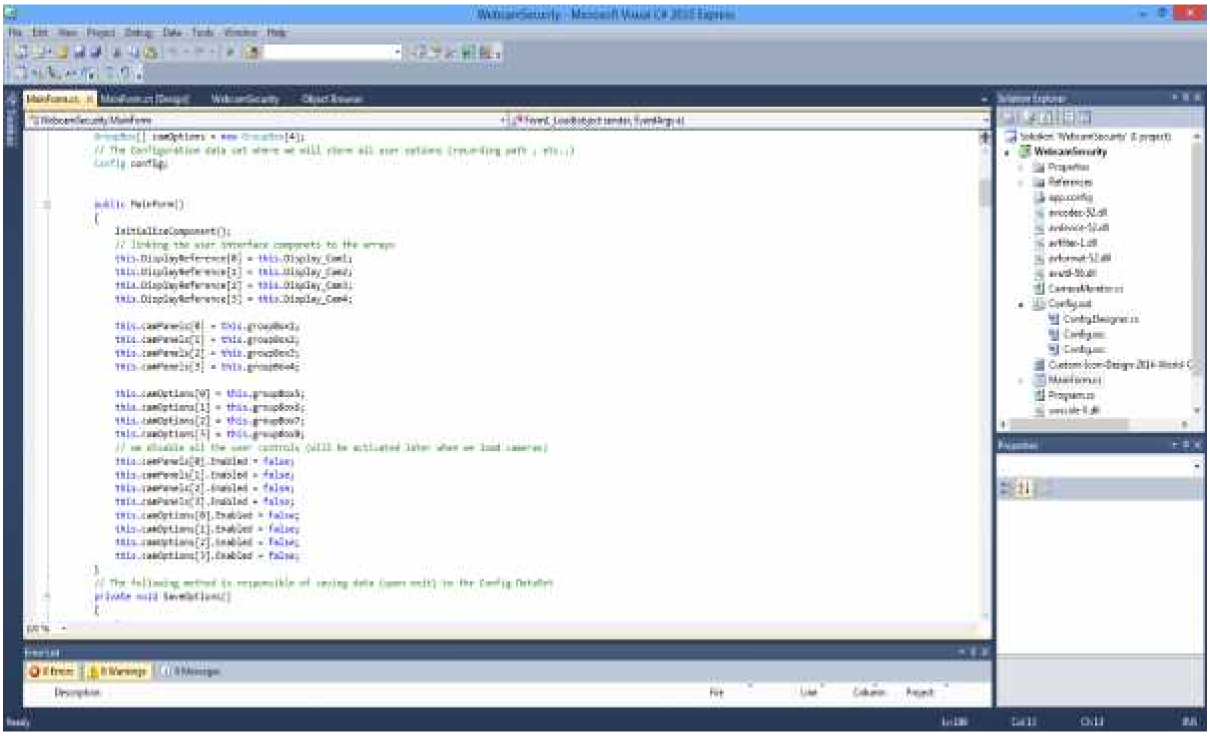
Text
The text associated with the control.

***Figure 1.2*:** Interface of main form in C#

3.4 SYSTEM IMPLEMENTATION

System Implementation is the next phase which involves nothing but writing software code and actually implementing the programming ideas and algorithms which have been designed or decided upon in the 'System Design' phase. Once the coding and implementation phase has been completed, it is now time to test the code.

Software code in C# based on our main form below.



```
GroupBox[] camOptions = new GroupBox[4];
// The Configuration data set where we will store all user options (recording path , etc..)
Config config;


public MainForm()
{
    InitializeComponent();
    // linking the user interface components to the arrays
    this.DisplayReference[0] = this.Display_Cam1;
    this.DisplayReference[1] = this.Display_Cam2;
    this.DisplayReference[2] = this.Display_Cam3;
    this.DisplayReference[3] = this.Display_Cam4;

    this.camPanels[0] = this.groupBox1;
    this.camPanels[1] = this.groupBox2;
    this.camPanels[2] = this.groupBox3;
    this.camPanels[3] = this.groupBox4;

    this.camOptions[0] = this.groupBox5;
    this.camOptions[1] = this.groupBox6;
    this.camOptions[2] = this.groupBox7;
    this.camOptions[3] = this.groupBox8;
    // we disable all the user controls (will be activated later when we load cameras)
    this.camPanels[0].Enabled = false;
    this.camPanels[1].Enabled = false;
    this.camPanels[2].Enabled = false;
    this.camPanels[3].Enabled = false;
    this.camOptions[0].Enabled = false;
    this.camOptions[1].Enabled = false;
    this.camOptions[2].Enabled = false;
    this.camOptions[3].Enabled = false;
}
// The following method is responsible of saving data (upon exit) to the Config DataSet
private void SaveOptions()
{
```

Figure 1.3: Source Code in C# for our main form

Our main programming language for our software development is done in C#. We chose C# based on the role of our end users, that are classified into two groups of concern namely "User Concern" and "Designer Concern". We will elaborate on these concerns more at the next phase of the Software Development Life Cycle.

3.5 SYSTEM MAINTENANCE

As the name suggests, the last phase is nothing but handing over the completed project to the client or customer, and subsequently performing maintenance activities on a periodic basis. We have outline four basic steps that will always guide us through our system maintenance procedures.

Software Maintenance Planning Procedures.

1. Adaptive – modifying the system to cope with changes in the software environment (OS) .
2. Perfective – implementing new or changed user requirements which concern functional enhancements to the software.
3. Corrective – diagnosing and fixing errors, possibly ones found by users.
4. Preventive – increasing software maintainability or reliability to prevent problems in the future.

3.6 JUSTIFICATION OF TOOLS

Microsoft Visual Studio C#

Microsoft Visual C# is Microsoft's implementation of the C# specification, included in the Microsoft Visual Studio suite of products. It is based on the ECMA/ISO specification of the C# language, which Microsoft also created. While multiple implementations of the specification exist, Visual C# is by far the one most commonly used. In most contexts, an unqualified reference to "C#" is taken to mean "Visual C#. Visual C# is currently used in development of Windows and Xbox Live games via Microsoft XNA, which allows game developers to create and share their games with other gamers. Visual C# is also heavily used by ASP.NET web sites and standalone applications based on the .NET Framework.

Aforge.Net

AForge.NET is an open source C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc.

The framework is comprised by the set of libraries and sample applications, which demonstrate their features:

AForge.Imaging - library with image processing routines and filters;

AForge.Vision - computer vision library;

AForge.Video - set of libraries for video processing;

AForge.Neuro - neural networks computation library;

AForge.Genetic - evolution programming library;

AForge.Fuzzy - fuzzy computations library;

AForge.Robotics - library providing support of some robotics kits;

AForge.MachineLearning - machine learning library; etc

Motion Detection

Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object.

The figure below (figure 1.3) gives a flow chart of how the motion detection works.

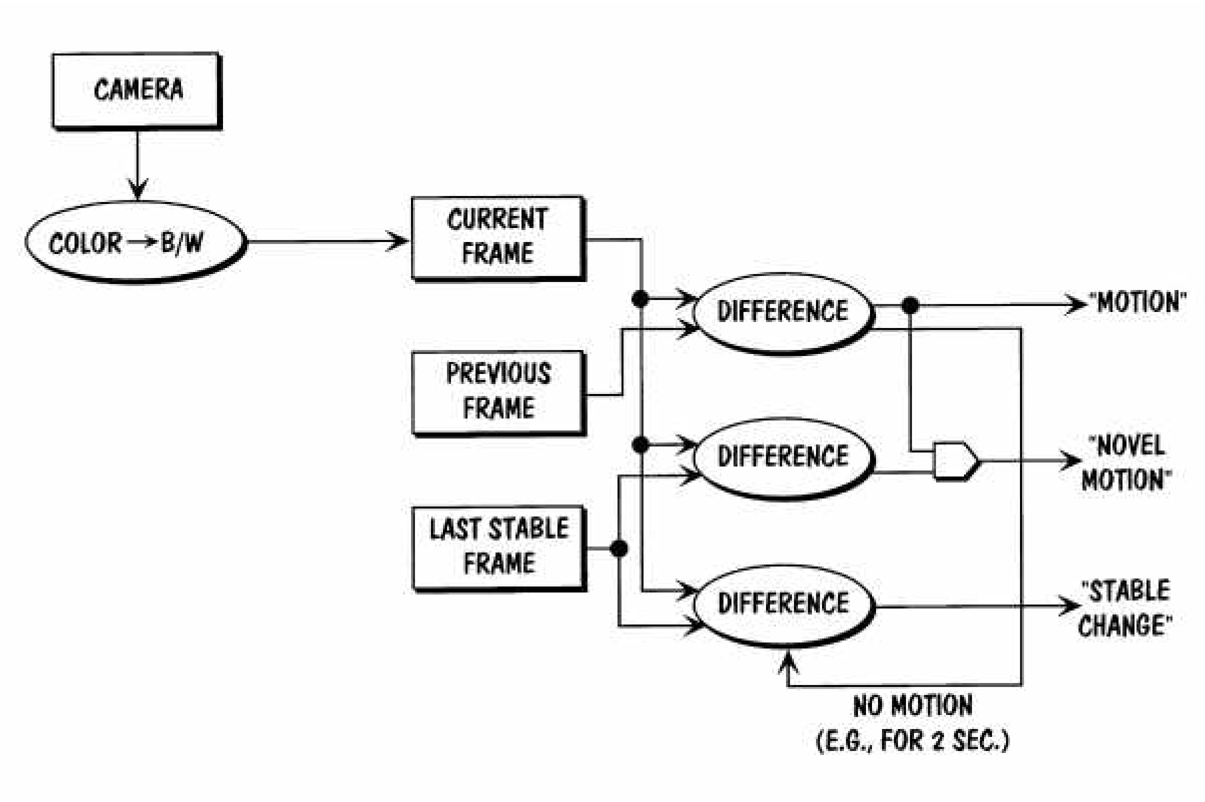


Figure 1.4: Flow Chart for Motion Detection

CHAPTER FOUR

TESTING

4.1 INTRODUCTION

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects). Software testing can be stated as the process of validating and verifying that a computer program/application/product:

- meets the requirements that guided its design and development,
- works as expected,
- can be implemented with the same characteristics,
- and satisfies the needs of end users.

Server Basic Requirements for SAMDAN 2.0

Our system is being design to run on every computer with the following;

- ✓ A minimum operating system of Windows XP
- ✓ Pentium IV(4) of 1GHz processor speed or more
- ✓ RAM of 512MB or more
- ✓ A Keyboard, Mouse and Webcam
- ✓ requires about 50MB of installation space.
- ✓ Extra storage device will be needed for systems video output

- ❖ But its highly recommended that you install a microsoft.net framework 3.5 on your server before the system can be able to work as expected.

Testing of the software was conducted in two phases-the alpha and beta testing. The Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization. Our alpha test was done when we developed the SamDan 1.0 (Alpha Version). To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing, i.e SamDan 2.0 (Beta Version).

The Beta test is the process of testing the product amongst Security personnel to confirm that the product works. The beta test consisted of real-time testing in a working environment. This was done by implementing and testing the software by Sarpnet Impression and some students of the Christian Service University College. Beta testing involves personal unstructured interviews which involve asking open-ended questions and also soliciting ideas and opinions from security personnel and officers.

4.2 SYSTEM TESTING

System Testing is the code that has been written is subjected to a series of tests and test cases to detect and determine whether there are any bugs, errors or software failures. Once all the repair work, i.e. correcting and re-writing every piece of erroneous or flawed code is completed. The testing is done to meet all your target requirements. Our target includes the following;

User concerns

1. Will the system deliver the information I need for my work?

2. How quickly can I access the data?

3. How easily can I retrieve the data?

4. How much clerical support will I need to configure the system?

5. How will the operation of the system fit into my daily business schedule?

Designer concerns

1. How much disk storage space will the master file consume?

2. How many lines of program code will it take to perform this function?

3. How can we cut down on CPU time when we run the system?

4. What are the most efficient ways of storing this data?

5. What video output format should our system use?

Now we've tested our software based on these questions and are really satisfied with the results. We then move to the next and last phase titled 'System Maintenance'.

4.3 SOFTWARE TESTING

The figure below (figure 2.1) shows the main icon in launching the beta version known as SamDan 2.1 by name SamDan Securities.

Figure 2.1: SamDan security icon

We chose the Ghana flag as our icon for the software of our project because we are happy and proud to be students of our motherland Ghana.

The figure below (figure 2.2) shows how the interface of the software when launched.

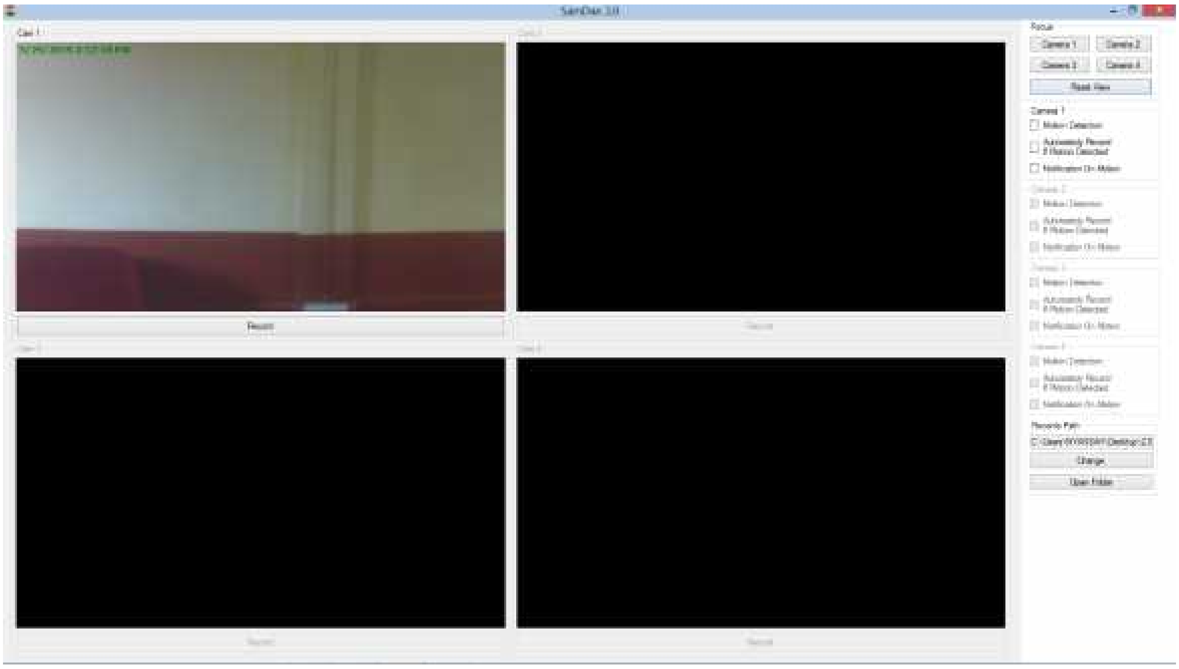


Figure 2.2: Main Interface of samdan 2.0

When the SamDan application is launched, it automatically detects any webcam that is connected to the server or computer system, as shown above (figure 2.2). Another unique feature about our software is that it has a time-stamped which is always active throughout the whole period. Figure 2.2 also shows that you can actually connect up to four webcams to the server in order to use all the four video displays on the software. And each cam has its own control buttons which are located on the right side of the software interface.

Figure 2.3will demonstrate to you how you can enlarge a video display to a full screen mode for a better view

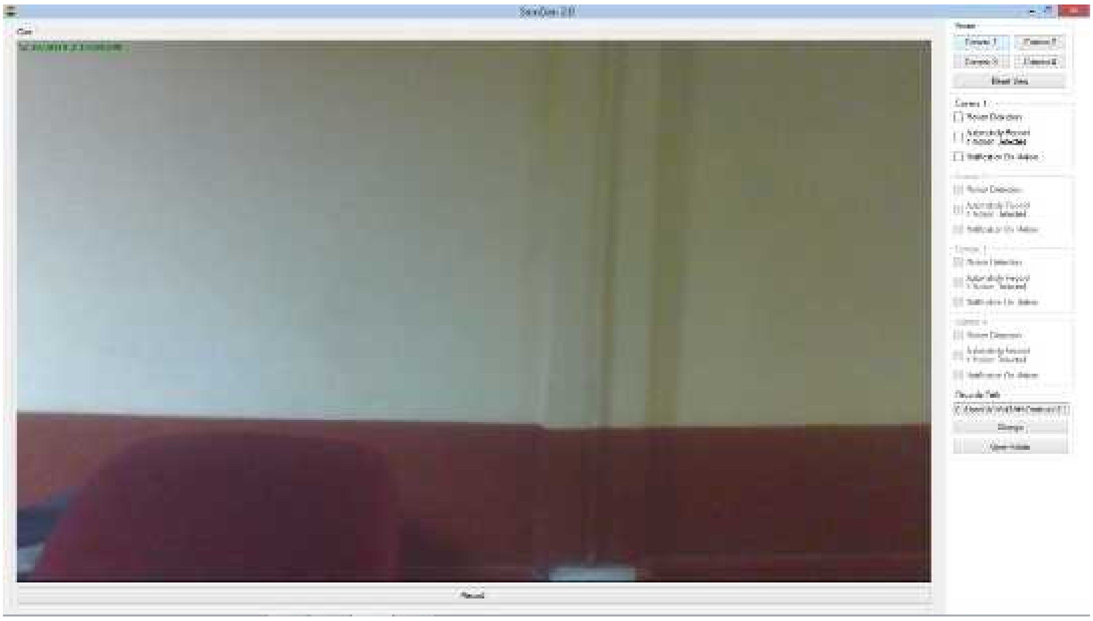


Figure 2.3: Enlarged video display of cam1

All you have to do is to click on any of the active webcam (e.g. Cam 1) and you will have you full screen view of that particular camera. In order to return back to the normal mode, there is a button named "Reset View" which will take you back to the default viewing mode.

Figure 2.4 shows you how to select the destination folder or output folder of where you want to store your video output files.

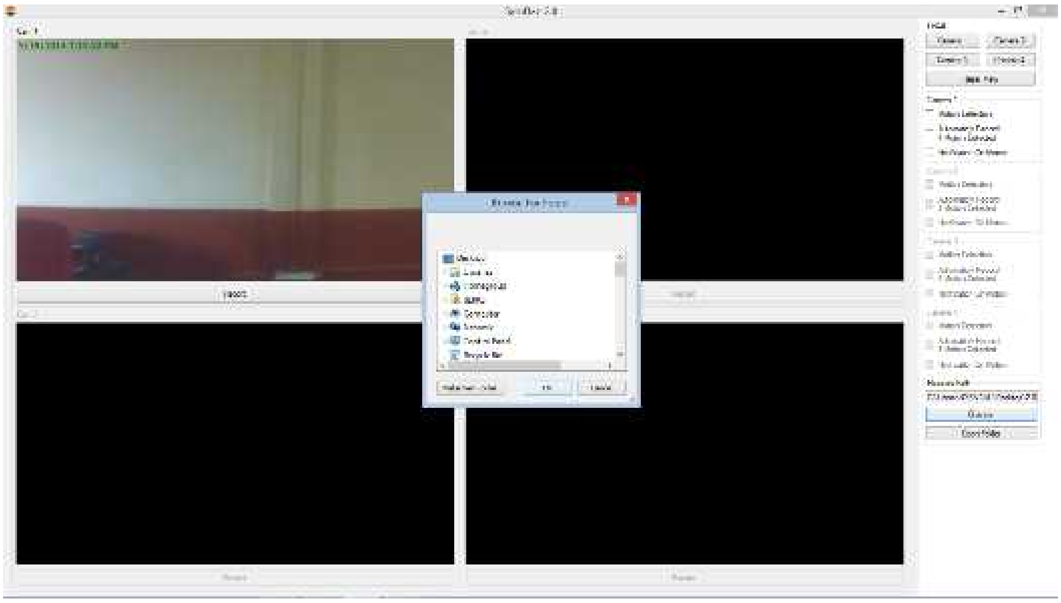


Figure 2.4: Output file destination

On the right bottom side of the interface is where you will find the output folder where you can decide to "**Change**" or "**Open Folder**" to view the recorded files. And it is very simple to navigate your way through.

The next figure (figure 2.5) shows how the output folder will look like when opened.

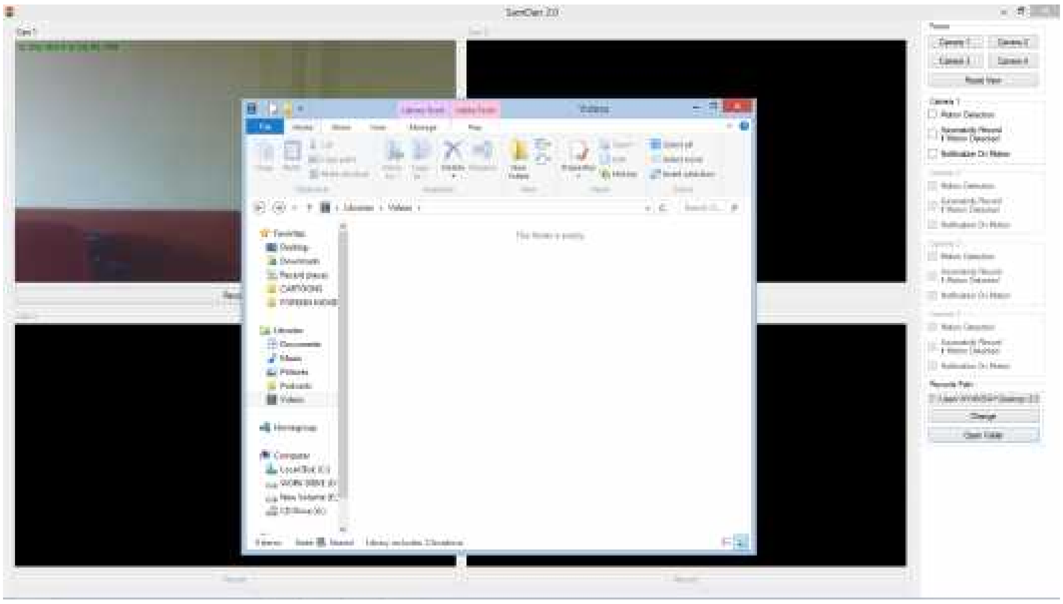
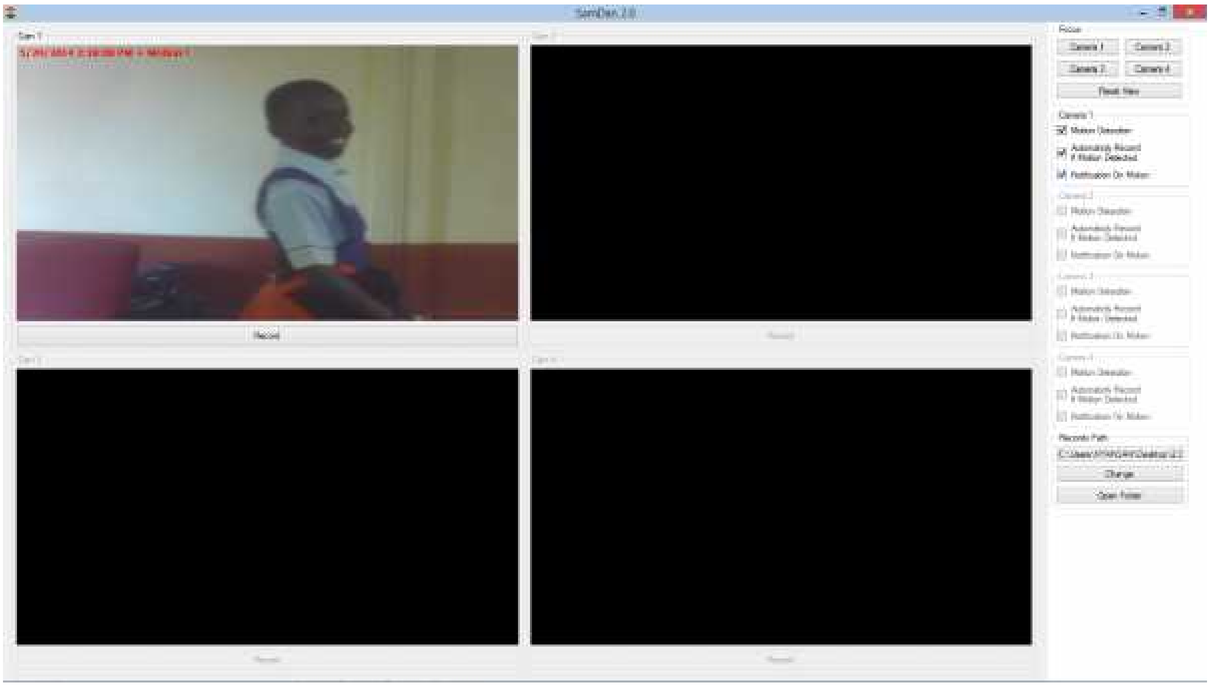


Figure 2.5: Output folder

Finally the next figure (figure 2.6) shows our most valuable and special key that makes our project unique. This is the motion detection mode we included in our software.



Camera 1

☑ Motion Detection

☑ Automatically Record if Motion Detected

☑ Notification On Motion

Camera 2

□ Motion Detection

□ Automatically Record if Motion Detected

□ Notification On Motion

Camera 3

□ Motion Detection

□ Automatically Record if Motion Detected

□ Notification On Motion

Camera 4

□ Motion Detection

□ Automatically Record if Motion Detected

□ Notification On Motion

Figure 2.6: Motion detected by software.

The motion detection works when you tick in the checkbox which are found on the software interface on the top right corner below the "**Reset View**" button. We have a "**Motion Detection**" which detects any movement within the camera's range. The second checkbox is "**Automatically Record if Motion is Detected**", and this will only be active when the "Motion Detection" has been selected. Thirdly we have the "**Notification on Motion**" which makes a beeping sound every 5seconds when a motion is being detected.

You can also see in figure 2.6 that some testing was made, where a girl who was acting as a test subject was detected by our security monitoring system.

The testing of our software has been successful and has met all our basic requirements.

CHAPTER FIVE

EVALUATION AND DOCUMENTATION

5.1 EVALUATION

A software evaluation is a type of assessment that seeks to determine if software or a combination of software programs is the best possible fit for the needs of a given client. The idea is to look closely at the resources and tools provided by the software that is either currently in use or is being examined as a possible addition to programs already in use by that client. There are several factors to consider with any software evaluation. One has to do with compatibility of the software with the hardware resources already in place on the client's network or computer equipment. Here, the focus is on the type of operating system the software requires in order to function, as well as the amount of memory and capacity that the hardware currently provides.

The table below shows an evaluation of our software done by some people and colleagues.

SamDan 2.0

	Not at all Important	Somewhat Unimportant	Moderately Important	Important	Extremely Important
Ease of Installation	○	○	○	○	◉
Hardware Compatibility	○	○	○	○	◉
Operating System Compatibility	○	○	○	○	◉
Consistency with Interface	○	○	○	◉	○
Application Software Compatibility	○	○	○	◉	○

5.2 DOCUMENTATION

According to (Gurdinson, 2010), we define documentation as any artifact that helps communicate information about the software systems and it is expected to provide precise information about software system.

Generally documentation artifacts respond to two main needs:

- it helps a software development team to communicate information about the software system and its implementation details to the maintenance team
- It aims to support software development team members to continuously formulate the solution to be implemented.

Usually documentation arises from the need to manage software projects' knowledge. The production of documentation is a way of making this knowledge explicit and available to others, at the present and future times. In earlier works, software documentation mainly refers to product manuals. Barker defined software documentation as —the design, planning, and implementation of any interface element, written and online, of a software system to enhance the system's usability. Based on this definition, documentation referred to software product manuals that were written for guiding users to use systems. In this terminology, documentation did not cover the technical documentation used across software development life cycle, e.g., requirements or design documents. (Golara Garousi, a Hybrid Methodology for Analyzing Software Documentation Quality and Usage, September 2012b)

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 CONCLUSION

SamDan Webcam Security Monitoring System is an advanced video surveillance software. Users can effortlessly monitor home, office, cradle, parking area, storehouse or any other premises 24-hours a day. Timestamped video capturing let users capture details of events precisely when they happen. Simply connect up to four(4) USB webcam to your PC and your good to go. Different environments have different surveillance requirements. A large facility like a parking lot, store, residence, or hall cannot be monitored efficiently by a single camera. Advances in PC based surveillance software now allow anyone with a webcam to setup a robust, effective and inexpensive surveillance system. Today, all you need for securing your assets is a PC, a couple of webcams and software like what we are designing to be used for multi camera monitoring.

6.1 RECOMMENDATIONS

The software is recommended for any household that wants a security system. Since it is mandatory for people to know how best you can monitor your safety. The software is also good for security systems monitoring which will detect intruders and the software can give an alarm notification.

- The main difficult task is that system has to be monitored form time to time in order to check for any intrusion updates..
- The software has the ability to record and stop recording when an intrusion is detected or not, so it needs to be placed in a secured room away from the children at home.

6.2 REFERENCES

Audrey, R. D., & Barbazy, C. A. (2008). The need of security systems. *Oxford Business Review*, *18*(6), 112-128.

Dougbus, N. (2010). Security guarding. Retrieved from http://securityguard.blogspot.com/

Frankies, N. and Zalakrish, H.(2011). Detecting Basic Security Processes. In *Proc. 2nd Symposium on Security Systems Design and Implementation*, Bronx, NY.

Gavin L. (2008) . A compiler for the Analysis of Security Protocols, Journal of Computer Security.

Georges G. M. & Ronald J. N. (2006). Unified Software Engineering with Java. p.201.

Hallawell, A. (2004). Re-evaluate the Privacy Risks of Hosting Data in the U.S. A published Gartner report.

Howell, K. E. (2013) Introduction to the Philosophy of Methodology. London: Sage Publications.

James, K.S (2009a): Basic Approach on the Effects of Home Security Systems

James, K.S (2009b): Basic Approach on the Effects of Home Security Systems

James, K.S (2009c) : Basic Approach on the Effects of Home Security Systems

Kuhn, D.L (1989). "Selecting and effectively using a computer aided software engineering tool". Annual Westinghouse computer symposium; 6-7 Nov 1989; Pittsburgh, PA (USA); DOE Project.

Loucopoulos P. and Karakostas V. (1995). System Requirements Engineering. McGraw-Hill.

Rackl, G. (2001). *Monitoring and managing heterogeneous middleware* (Doctoral dissertation, Technische Universität München, Universitätsbibliothek).

Rasmussen, S. (2002). Centralized Network Security Management, Combining Defense in Depth with Manageable Security, SANS Institute Security Reading Room, Retrieved May, 2014, from http://www.sans.org/ reading room/).

Robinson K. (1992). Putting the Software Engineering into CASE. New York : John Wiley and Sons Inc.

Ryan P. , Schneider S. & Goldsmith M. (2001). " Modelling and Analysis of security Protocols.

Schneir, B. (2005a), *Managed Security Monitoring*: Network Security for the 21st Century.

Schneir, B. (2005b) *Managed Security Monitoring*: Network Security for the 21st Century.

Wang, H., Wang, H. H., & Wang, H. H. (1999). *Telecommunications Network Management*. McGraw-Hill Professional.

Webopedia (2013). Software security: http://en.wikipedia.org/wiki/Software. (accessed on 01/11/13)

Yusuff, A.(2013a): Network monitoring, Bachelor’s thesis Central Ostrobothnia University of applied Sciences Degree Programme in Information Technology

Yusuff, A. (2013b): Network monitoring, Bachelor’s thesis Central Ostrobothnia University of applied Sciences Degree Programme in Information Technology .

Xiao He (2007). "A metamodel for the notation of graphical modeling languages". In: Computer Software and Applications Conference, 2007. COMPSAC 2007 - Vol. 1. 31st Annual International, Volume 1, Issue , 24–27 July 2007, pp 219-224.