VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

"Smart Surveillance System"

Thesis submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by

1CR14CS011	Amrit Sinha
1CR14CS015	Ankur Singh
1CR14CS076	Manas Kashyap

Under the Guidance of

Mrs. Swetha K V
Assistant Professor
Department of CSE, CMRIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Certificate

This is to certify that the project entitled **"Smart Surveillance System"** is a bonafide work carried out by **Amrit Sinha (USN:1CR14CS011), Ankur Singh (USN:1CR14CS015), and Manas Kashyap(USN:1CR14CS076)** in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

----------	----------	----------
Signature of Guide	Signature of HoD	Signature of Principal
Mrs. Swetha K V	**Dr. Jhansi Rani P**	**Dr. Sanjay Jain**
Assistant Professor	Professor & Head	Principal
Department of CSE	Department of CSE	CMRIT,
CMRIT, Bengaluru - 37	CMRIT, Bengaluru - 37	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	----------	----------	----------
2.	----------	----------	----------

Acknowledgement

"**Acknowledgement** - We take this opportunity to thank all of those who have generously helped us to give a proper shape to our work and complete our BE project successfully. A successful project is fruitful culmination efforts by many people, some directly involved and some others indirectly, by providing support and encouragement."

We would like to thank **Dr.SANJAY JAIN**, Principal, CMRIT, for providing excellent academic environment in the college.

We would like to express our gratitude towards **Dr.JHANSI RANI**, Professor & HOD, Dept of CSE, CMRIT, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our Internal Guide **Mrs.SWETHA K V**, Asst. Professor, Department of Computer Science & Engineering, CMRIT, for her valuable guidance throughout the tenure of this project work.

Amrit Sinha
Ankur Singh
Manas Kashyap

Table of Contents

List of Figures

List of Tables

Abstract

Nowadays the need for a safe and secure system is desired by each and every individual in the society. The most commonly used system, Closed Circuit TeleVision (CCTV) is being implemented everywhere such as in hospitals, warehouses, parking lots, buildings etc... However this very system though effective has its downside when it comes to cost. Thus the need for a cost effective system is required. In this project we propose to use a security camera with the night vision capabilities using raspberry pi and openCV. This is a cost effective method that uses a credit card sized chip RPI. The image is captured and each frame is processed. The image is stored and an email is sent if human is detected. This system has accuracy of about 83 %. Also we use a pi camera. So the image is captured via the pi camera and it is send to the raspberry pi for processing for face and human detection with the help of openCV. Then, the face detected is compared with the database, if the human detected is known (visitor) or not (stranger) and based on the output, an email is generated and is sent to the user.The security camera can be controlled using a custom AI assistant. The AI features include turning on and turning off the security camera using voice control. Thus, one can provide a low cost security system.

Chapter 1

Preamble

1.1 Introduction

Nowadays, people want one sole thing that is to make them feel safe and secure. The most commonly used security system is the CCTV (closed circuitTelevision). The cost of implementation of CCTV varies depending upon the size and use of the system. It is usually installed in hospitals, malls, parking lots etc. However, with the help of CCTV one can monitor the area 24/7, or the footage if stored in a location can beretrieved when required. Although, it can be used to deter crime and allows the authorities to identify and solve a crime, it doesn't detect neither recognize the person who is involved.

We have implemented a system which provides both face detection and face recognition with the help of Raspberry pi 3 which is a credit card sized minicomputer and a Pi camera which is made especially for the raspberry pi 3. Thus, when dealing with the real-time image processing, Open source computer vision(openCV) software,a powerful library of image processing tools, is a good choice. With the help of a smart surveillance system, we have achieved a system that can record the event, detect and recognize the person. A GSM module is used to send a message stating whether the person is an intruder or a visitor. If it is a visitor, then a command is sent by the user to perform some operation like- open the door (any type of automation is implemented) however if it is a stranger an alarm is generated to indicate that there is an intruder.

1.2 Problem Statement

It is necessary to make use of automatic video analysis technologies for developing smart surveillance system which can aid the human operator in both detecting and reacting to potential threats.The internet and wireless broadband infrastructure is becoming robust enough to permit excellent remote video surveillance. With advances in hardware and software technology, and the emergence of ubiquitous internet infrastructure and wireless networks with broadband capability, it is now possible to design and build a networked video surveillance system that can do an excellent job of remote video supervision from anywhere and at any time. The requirements of a video surveillance system differs in important ways from CCTV, NVR's and DVR's. Smart Video Surveillance System(SVSS) provides video based object analysis capabilities.

The developed SVSS provides a wide range of features in order to solve the following problems in surveillance areas:

- Detecting objects.
- Face detection.
- Sending prompt messages in case of intruder.
- Internet Of Things(IOT)
- AI Assistant

Chapter 2

Literature Survey

2.1 Introduction

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

2.2 Literature Survey

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary sources data in the areas of specific interest to the researcher. The library is a rich storage base for secondary data and researchers used to spend several weeks and sometimes months going through books, journals, newspapers, magazines, conference proceedings, doctoral dissertations, masters theses, government publications and financial reports to find information on their research topic. Reviewing the literature on the topic area at this time helps the research er to focus further interviews more meaningfully on certain aspects found to be important is the published studies even if these had not surfaced during the earlier questioning .So the literature survey is important for gathering the secondary data for the research which might be proved very helpful in the research. The literature survey can be conducted for several reasons. The literature review can be in any area of the business.

2.2.1 Paper 1

SMART SURVEILLANCE CAMERA USING RASPBERRY PI 2 AND OPENCV

Context: Abstract Nowadays the need for a safe and secure system is desired by each and every individual in the society. The most commonly used system, Closed Circuit TeleVision (CCTV) is being implemented everywhere such as in hospitals, warehouses, parking lots, buildings etc... However this very system though effective has its downside when it comes to cost. Thus the need for a cost effective system is required. The existing system for surveillance is a security camera with the night vision capabilities using raspberry pi and openCV. This is a cost effective method that uses a credit card sized chip RPI. The image is captured and each frame is processed. The image is stored and an email is sent if human is detected. The existing system has accuracy of about 83 %. In this project we propose to use an enhanced recent model- raspberry pi 2 which has operating speed 900MHz. Also we use a pi camera. So the image is captured via the pi camera and it is send to the raspberry pi 2 for

processing for face and human detection with the help of openCV. Then, the face detected is compared with the database, if the human detected is known (visitor) or not (stranger) and based on the output, an audio output is produced and a message is sent to the user. Thus, one can provide a low cost security system.

2.2.2 Paper 2

Low Cost Smart Security Camera with Night Vision Capability Using Raspberry Pi and PIR Sensor

Context:Abstract: In order to further maintain peace and provide security to people now a day, Closed-circuit television (CCTV) surveillance system is being utilized. This study focused on the design and implementation of a low cost smart security camera with night vision capability using Raspberry Pi (RPI) with PIR. The system was designed to be used inside a warehouse facility. It has human detection and smoke detection capability that can provide precaution to potential crimes and potential fire. The credit card size Raspberry Pi (RPI) with A passive infrared sensor (PIR sensor) handles the moving body, control algorithms for the alarms and sends captured pictures to users email via Bluetooth. As part of its alarm system, it will play the E-speech sounds: intruder when there is detection. The system uses ordinary webcam but its IR filter was removed in order to have night vision capability. With help of LDR it will sense whether it is night or day if it is night the led will on when it detect intruder.

2.2.3 Paper 3

Motion sensor and face recognition based surveillance system Using Raspberry Pi

Context: An intelligent surveillance system is a smart monitoring system which is developed from the security point of view. The objective of this project is to develop a system that monitors the area in which it is being implemented. An Intelligent surveillance system is applicable in the area where no one is permissible to enter, also where we need to detect if any motion has been done. For this a digital camera is used. By combining the PIR sensor and camera we can use this system to detect the motion. The Camera is used to catch the live images of the area in which it is being implemented, if any object is moving. The captured images are stored in a particular folder. The stored images will be then useful to work on.. As the PIR sensor recognize the motion it uses local binary pattern histogram (LBPH) and matches his/her face with the data provided in the database. If the face is not matched with the database then he is unauthorised and the buzzer starts buzz ring including the message/mail services to the owners.

2.2.4 Paper 4

Object Detection on Raspberry Pi

Context: Object dis covery and following are es s ential and tes t ing undertakings in numerous PC v is ion applications , for e xa m ple, reconnaiss ance, vehicle route and s elf-s ufficient robot route. Ques tion recognition includes finding objects in the cas ing of a video s uccess ion. Each following strategy requires a question recognition component either in each edge or when the protest firs t s hows up in the video. For ins tance figuring the meas ure of a planet(as tronomy), d is tinguis hing dis eas e in a ma mmograp hy s can(medicine), abs taining fro m controlling into an obstacle(robotics), and identify ing a man's eye s hading or hair co lor(s ec urit y). The goal is to construct a model that can recognize the protes t of indicated s hading that make utilization of open s ource equipment and that chips away at the premis e of vis ual informat ion caught from an ordinary webcam which has a reasonable lucidity. Having a picture p reparing calculat ion which dis tinguis hes a protest firs t and after that tracks it the length of it is in the observable pathway of the camera. As the protes t moves , thePC/portable PC/implanted Board offers flag to engine to turn the came ra which is mounted on a stepper engine. We executed the proposed calculation on Raspberry Pi board utilizing OpenCV on Linu x foundation. It is a proficient protes t following technique which take ca re of the fo llo wing is s ues .

2.2.5 Paper 5

Smart Surveillance System using Raspberry Pi and Face Recognition

Context: Abstract: This paper proposes the Smart Surveillance System using Raspberry Pi and Image Processing. This system will serve as smart security module for monitoring. Traditional surveillance systems only records the activities based on motion, but this system serves the purpose of facial recognition so as to reduce the error caused due to motion detection .Raspberry Pi camera module is used to capture images once the motion is detected by the PIR Sensor. This system will monitor when motion detected and checks for the faces in the image captured and with the help of face recognition alerts if the face detected is not stored in the database.

2.2.6 Paper 6

Survey Paper on Smart Surveillance System

Context: This paper deals with the survey of Smart surveillance monitoring system using Raspberry pi. Video Surveillance is important as far as security is concerned these days. Commercial spaces, schools and hospitals, warehouses and other challenging indoor and outdoor environments require high end cameras. The current

technologies require RFIDs which are costly and hence the security domain in all becomes expensive and hence there was a need to work on this. This paper describes the use of low cost single on board computer Raspberry Pi. This new technology is less expensive and in this project it is used as a standalone platform for image processing. It increases the usage of mobile technology to provide essential security to our homes and for other control applications. The proposed home security system captures information and transmits it via a 3G Dongle to a Smart Phone using web application Raspberry pi.

2.2.7 Paper 7

Enhanced Home Security Using IOT and Raspberry PI

Context: A smart home application features great help to our everyday life. This system rejuvenates facilities of a house to evolve into a smart home by adding more security features. The improvement in security aspect offers innovative and productive scope to the means of living. All these characteristics is adapted by using Internet of Things (IoT) and Raspberry Pi. The recognition problem is always questionable in smart home applications. So, a recovery is done to identify the intruder as known or unknown by the use of image processing techniques for face recognition. This tend to solve many issues in terms of authentication. This protection mechanism notifies the user accordingly, giving a clear picture of the scenario happening at the users house. The sensor based system highlights many features enabling it to be widely used. Fire sensor detects any temperature increase in the living room and posts its status in the URL given to the user. The gas sensor helps in detecting the presence of any gas leakage based on the intensity of the gas in air. With the help of DC motor, auto door locking mechanism is actuated. This is very useful. All the statuses are processed between the sensors and the user via IoT. Raspberry Pi connects all the components and brings forth the proper functioning of the whole package. The procedures that are used here are very simple. Hence, even novice users could understand the systems advanced features and use it with ease. The use of surveillance camera also helps in identifying the presence of flame and thus a buzzer is activated in the case of fire detection.

2.2.8 Paper 8

Real Time Face Detection and Tracking Using OpenCV

Context: In this paper, we intend to Implement a real-time Face detection and tracking the head poses position from high definition video using Haar Classifier through Raspberry Pi BCM2835 CPU processor which is a combination of SoC with

GPU based Architecture. SimpleCV and OpenCV libraries are used for face detection and tracking the head poses position. The experimental result computed by using computer vision SimpleCV and OpenCV framework libraries along with above mentioned hardware results were obtained through of 30 fps under 1080p resolutions for higher accuracy and speediness for face dete ction and tracking the head poses position.

Chapter 3

Theoretical Background

3.1 Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.[5][6][7] The original model became far more popular than anticipated,[8] selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards, mice and cases). However, some accessories have been included in several official and unofficial bundles.[8]

Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. The boards have one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. Lower-level output is provided by a number of GPIO pins which support common protocols like IC. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth. Prices range from US$5 to $35.

The first generation (Raspberry Pi 1 Model B) was released in February 2012, followed by the simpler and cheaper Model A. In 2014, the Foundation released a board with an improved design, Raspberry Pi 1 Model B+. These boards are approximately credit-card sized and represent the standard mainline form-factor. Improved A+ and B+ models were released a year later. A "Compute Module" was released in April 2014 for embedded applications. The Raspberry Pi 2 which added more RAM was released in February 2015.

A Raspberry Pi Zero with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015 for US$5. By 2017, it became the newest mainline Raspberry Pi. On 28 February 2017, the Raspberry Pi Zero W was launched, a version of the Zero with Wi-Fi and Bluetooth capabilities, for US$10.[15][16] On 12 January 2018, the Raspberry Pi Zero WH was launched, the same version of the Zero W with pre-soldered GPIO headers.[17]

Raspberry Pi 3 Model B was released in February 2016 with a 64 bit quad core processor, and has on-board WiFi, Bluetooth and USB boot capabilities.[18] On Pi Day 2018 model 3B+ appeared with a faster 1.4 GHz processor and a 3 times faster network based on gigabit ethernet (300 Mbit / s) or 2.4 / 5 GHz dual-band Wi-Fi (100 Mbit / s).[1] Other options are: Power over Ethernet (PoE), USB boot and network boot (an SD card is no longer required). This allows the use of the Pi in hard-to-reach places (possibly without electricity).

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

The Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialised media centre distributions.[19] It promotes Python and Scratch as the main programming language, with support for many other languages.[20] The default firmware is closed source, while an unofficial open source is available

3.2 Why are we using Raspberry instead of Arduino?

- Processing power: Most of the Arduino boards use a micro-controller which have low processing power as compared to the processors that are used by raspberry pi.
- Memory: Arduino boards have just enough memory to store your programs whereas raspberry pi has memory card slot which can be used for primary data storage just like the HDD of a PC or laptop.
- Networking: Raspberry pi 3 has both, a built in Wireless N connectivity and Ethernet port whereas there is no such technology that comes with Arduino directly out of the box. Although there are various shields and modules available that can help you with networking in Arduino.
- Sensors: Connecting analog sensors to Arduino is much easier as it can easily interpret and respond to a wide range of sensors whereas raspberry pi requires software to interface with such devices.
- VERDICT: If you are developing simple project that does not require networking and multitasking you should go for an Arduino, whereas if your project requires multitasking,networking and more complex computer functions you should go for a Raspberry pi.

3.3 What model of raspberry pi are we using?

For our project we are using Raspberry Pi 3 Model B

Figure 3.1: Raspberry Pi 3 Model b

3.3.1 WHAT IS THE RASPBERRY PI 3?

The Raspberry Pi 3 Model B is the latest version of the 35 dollars Raspberry Pi computer. The Pi isn't like your typical machine, in its cheapest form it doesn't have a case, and is simply a credit-card sized electronic board-of the type you might find inside a PC or laptop but much smaller.

3.3.2 WHAT IS THE RASPBERRY PI 3 CAPABLE OF?

A surprising amount. We can use the Pi 3 as a budget desktop, media center, retro games console, or router for starters. However that is just the tip of the iceberg. There are hundreds of projects out there, where people have used the Pi to build tablets, laptops, phones, robots, smart mirrors, to take pictures on the edge of space, to run experiments on the International Space Station – and that's without mentioning the wackier creations

3.3.3 HOW DO I GET STARTED WITH THE RASPBERRY PI 3?

One thing to bear in mind is that the Pi by itself is just a bare board. You'll also need a power supply, a monitor or TV, leads to connect to the monitor–typically HDMI, and a mouse and keyboard.

Once you've plugged in all the cables, the easiest way for new users to get up and running on the Pi is to download the NOOBS (New Out-Of-Box Software) installer. Once the download is complete, follow the instructions here and here and it will walk you through how to install an OS on the Pi. The installer makes it simple to set up various operating systems, although a good choice for first time users is the official OS Raspbian–although other operating systems are listed below.

The look and feel of Raspbian should be familiar to any desktop computer user. The OS, which is constantly being improved, recently had a graphical overhaul, and includes an optimized web browser, an office suite, programming tools, educational games, and other software.

3.3.4 HOW IS THE RASPBERRY PI 3 DIFFERENT FROM ITS PREDECESSORS?

The quad-core Raspberry Pi 3 is both faster and more capable than its predecessor, the Raspberry Pi 2. For those interested in benchmarks, the Pi 3's CPU–the board's main processor–has roughly 50-60 percent better performance in 32-bit mode than that of the Pi 2, and is 10x faster than the original single-core Raspberry Pi (based on a multi-threaded CPU benchmark in SysBench). Compared to the original Pi, real-world applications will see a performance increase of between 2.5x–for single-threaded applications–and more than 20x–when video playback is accelerated by the chip's NEON engine.

Unlike its predecessor, the new board is capable of playing 1080p MP4 video at 60 frames per second (with a bitrate of about 5400Kbps), boosting the Pi's media center credentials. That's not to say, however, that all video will playback this smoothly, with performance dependent on the source video, the player used and bitrate.

The Pi 3 also supports wireless internet out of the box, with built-in Wi-Fi and Bluetooth.

The latest board can also boot directly from a USB-attached hard drive or pen drive, as well as supporting booting from a network-attached file system, using PXE, which is useful for remotely updating a Pi and for sharing an operating system image between multiple machines.

3.3.5 WHICH OPERATING SYSTEMS CAN I RUN ON THE PI?

The Pi can run the official Raspbian OS, Ubuntu Mate, Snappy Ubuntu Core, the Kodi-based media centers OSMC and LibreElec, the non-Linux based Risc OS (one for fans of 1990s Acorn computers). It can also run Windows 10 IoT Core, which is very different to the desktop version of Windows, as mentioned below.

However, these are just the officially recommended operating systems, and a large array of other weird and wonderful OSes also work on the Pi.

3.3.6 HOW CAN I GET THE MOST FROM MY RASPBERRY PI 3?

It's good advice to get a case to protect the Pi from damage, especially if you're going to be carrying the Pi around with you.

If performance is important to you, you can also invest in a high-speed micro SD card, as outlined below.

While the Pi can run a lot of different operating systems, if you're after stability and performance then the official Raspbian operating system is a good choice, having been tuned to get the most from the Pi, and bundling a fast web browser and a decent selection of office and programming software.

If you didn't install the Raspbian OS using the Noobs installer, and you're running out of space, you can also go into the terminal and type 'sudo raspi-config' and then select the option to 'Expand root partition to fill SD card', which will ensure you're using all available space on the card.

3.3.7 WHAT IS THE POWER CONSUMPTION OF THE RASPBERRY PI 3?

According to tests, the peak power consumption of the Pi 3 when under heavy load is about twice that of the Pi 2 (750mA vs 360mA), though for less-demanding workloads it should be broadly similar to earlier boards.

3.3.8 WHAT POWER SUPPLY DO I NEED FOR THE RASPBERRY PI 3?

The best choice is the official Raspberry Pi Foundation power supply, which is rated at 2.5A5.1V. This is in contrast to the 2A5V-rated supply used by earlier boards.

3.3.9 WHICH IS THE FASTEST MICRO SD CARD FOR THE RASPBERRY PI 3?

A particularly fast card in a recent round-up was found to be the SanDisk Extreme PLUS 64GB microSDXC. For most users a standard class 4 micro SD card working at 4MB/s should suffice.

3.3.10 WHAT SIZE MICRO SD CARD DO I NEED FOR THE RASPBERRY PI 3?

If we are installing the official Raspbian OS you'll need at least an 8GB micro SD card, whereas for the Raspbian Lite you'll need a minimum of 4GB.

3.3.11 CAN I USE WI-FI ON THE RASPBERRY PI 3?

Yes, the board supports 802.11n Wireless LAN (peak throughput of 150Mbps) and Bluetooth 4.1.

3.3.12 CAN I RUN A NETWORK OF RASPBERRY PI 3S?

Yes, and managing and updating the boards should be made simpler by the ability to boot from a network-attached file system using PXE, allowing admins to share operating system images between machines.

3.4 Pi Camera Module

The Raspberry Pi Camera v2 is the new official camera board released by the Raspberry Pi Foundation.

The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 imagesensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens.

The Raspberry Pi Zero now comes complete with a camera port! Using the new Raspberry Pi Zero Camera Adapter, you can now use a Raspberry Pi camera to your Zero

It'scapable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and640x480p90 video.

It attaches to Pi by way of one of the small sockets on the board upper surface and uses thededicated CSi interface, designed especially for interfacing to cameras.

Features:

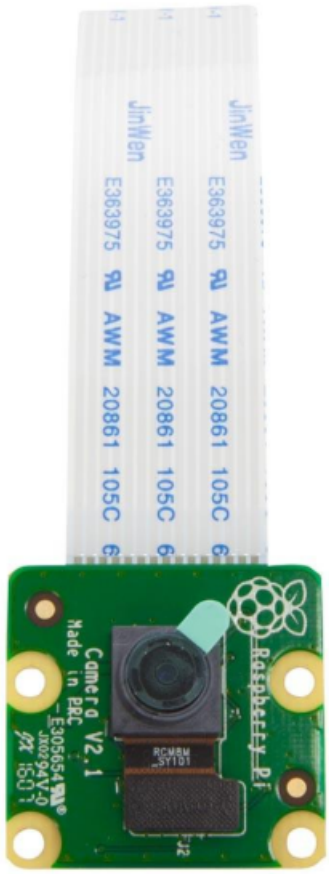

Figure 3.2: Raspberry Pi Camera Module

- Fixed focus lens on-board
- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Connects to the Raspberry Pi board via a short ribbon cable (supplied)
- Weight just over 3g
- Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferredoperating system

3.5 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common

infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCVs deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.6 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted Python is processed at runtime by the interpreter. You

do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features Python's features include

- Easy-to-learn Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain Python's source code is fairly easy-to-maintain.
- A broad standard library Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases Python provides interfaces to all major commercial databases.

- GUI Programming Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.7 Amazon Web Services

AWS was one of the first companies to introduce a pay-as-you-go cloud computing model that scales to provide users with compute, storage or throughput as needed. AWS launched in 2006 from the internal infrastructure that Amazon.com built to handle its online retail operations. AWS was one of the first companies to introduce a pay-as-you-go cloud computing model that scales to provide users with compute, storage or throughput as needed. Amazon Web Services provides services from dozens of data centers spread across availability zones (AZs) in regions across the world. An AZ represents a location that typically contains multiple physical data centers, while a region is a collection of AZs in geographic proximity connected by low-latency network links. An AWS customer can spin up virtual machines (VMs) and replicate data in different AZs to achieve a highly reliable infrastructure that is resistant to failures of individual servers or an entire data center.

3.7.1 Characteristics

1. On-demand self-service Users are able to provision cloud computing resources without requiring human interaction, mostly done though a web-based self-service portal (management console).

2. Broad network access Cloud computing resources are accessible over the network, supporting heterogeneous client platforms such as mobile devices and workstations.

3. Resource pooling Service multiple customers from the same physical resources, by securely separating the resources on logical level.

4. Rapid elasticity Resources are provisioned and released on-demand and/or automated based on triggers or parameters. This will make sure your application will have exactly the capacity it needs at any point of time.

5. Measured service Resource usage are monitored, measured, and reported (billed) transparently based on utilization. In short, pay for use.

Chapter 4

System Requirement Specification

4.1 Introduction

This chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of dissertation as well as the functional and non-functional requirement of this dissertation. A SRS document describes all data, functional and behavioral requirements of the software under production or development. SRS is a fundamental document, which forms the foundation of the software development process. Its the complete description of the behavior of a system to be developed. It not only lists the requirements of a system but also has a description of its major feature. Requirement Analysis in system engineering and software engineering encompasses those tasks that go into determining the need or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement Analysis is critical to the success to a development project. Requirement must be documented, measurable, testable, related to in identified business needs or opportunities, and defined to a level of detail sufficient for system design. The SRS functions as a blueprint for completing a project. The SRS is often referred to as the parent document because all subsequent project management documents, such as design specifications, statements of work, software architecture specification, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only. Thus the goal of preparing the SRS document is to

- To facilitate communication between the customer, analyst, system developers, maintainers.
- To serve as a contrast between purchaser and supplier.
- To firm foundation for the design phase.
- Support system testing facilities.
- Support project management and control.
- Controlling the evolution of the system.

4.2 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include

calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- Aquiring object images
- Aquiring face images
- Processing face images
- Face detection
- Automation

4.3 Non-functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non- functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do requirement, an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, nonfunctional requirements are in the form of system shall be requirement, an overall property of the system as a whole or of a particular aspect and not a specific function.

The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements of our project include:

- Security
- Maintainability - As a tool to obtain the ease of maintainability UML will be used in the development process.
- Portability - To ensure portability, the system will be developed in PYTHON language.

4.3.1 Product Requirements

- Portability: Since the SVSS system is designed to run using Raspberry Pi (whose library is written in Python), the system is portable.
- Correctness: It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.
- Ease of Use: The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.
- Modularity: The complete product is broken up into many modules and welldefined interfaces are developed to explore the benefit of flexibility of the product.
- Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

whereas evolution quality involves testability, maintainability, extensibility or scalability

4.3.2 Organizational Requirements

Process Standards: IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world. Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.4 User Requirements

The user requirements document (URD) or user requirements specification is a document usually used to software engineering that specifies the requirements the user expects from software to be constructed in a software project. Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand feature not in the URD, whilst the developer cannot claim the product is ready if it does not meet an item of the URD. The URD can be used as a guide to planning cost, timetables, milestones, testing etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described. Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lies between a science and economically feasible. Preparing a URD is one of those skills that lies between a science and an art, requiring both software technical skills and interpersonal skills.

4.4.1 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request. Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets. Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them. An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

- Where are we now?
- Where do we want to be?
- How do we get there?

The customers are those that perform the eight primary functions of systems engineering,with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.
- Performance and related parameters: It points out the critical system parameters to accomplish the mission
- Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- Operational life cycle: It defines the system lifetime

4.4.2 Hardware requirements

- Micro-processor: Raspberry Pi 3
- SoC: Broadcom BCM2837
- CPU: 4 ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: microSD
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4 USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Camera: Raspberry Pi Camera Module
- Operating System: Raspbian

4.4.3 Software Requirements

- Operating System: Linux
- Coding Language: Python
- Library: OpenCV

4.4.4 Software Quality Attributes

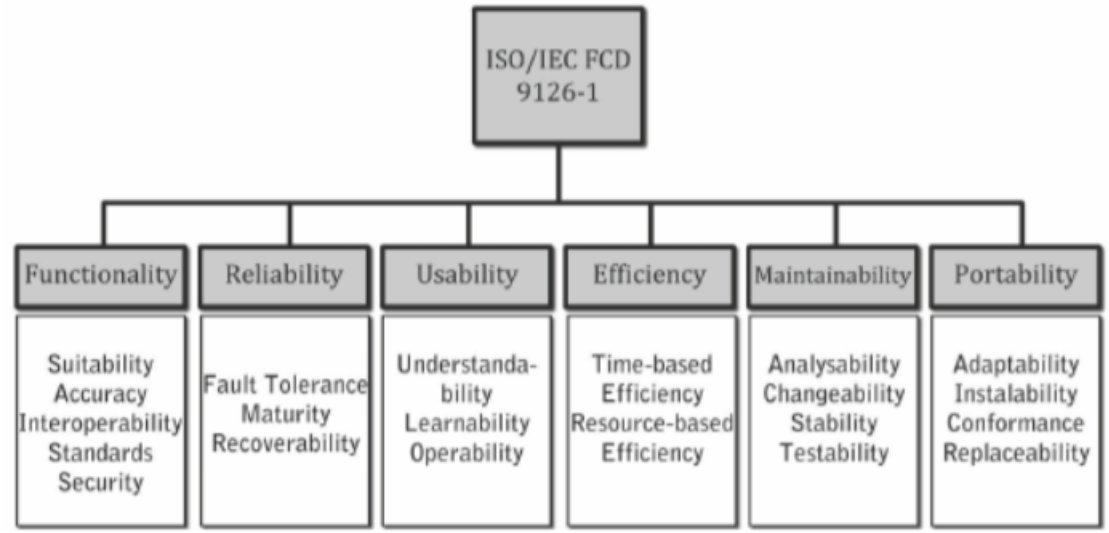

Figure 4.1: Software Quality Attribute

- Functionality: the capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.
- Reliability: the capability of the software to maintain its level of performance when used under specified conditions.
- Usability: the capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.
- Efficiency: the capability of the software to provide the required performance, relative to the amount of resources used, under stated conditions.
- Maintainability: the capability of the software to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

- Portability: the capability of software to be transferred from one environment to another.

Chapter 5

System Analysis

5.1 Introduction

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

5.2 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility
- Social Feasibility

5.2.1 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirement analysis phase of system development. As the operation of the existing signature pads and their relevant softwares are user friendly and do not require any expertise, the proposed system also supports user friendly approach and less necessity for expertise in technology and operation of the system. In general, the proposed system relies on the fact that creating a forgery indistinguishable from a genuine signature requires expert knowledge, that mistakes

are easily made, or that the amount of effort required to do so is considerably greater than the amount of money that can be gained.

The feasibility of implementing a face recognition system in an FPGA device based on Gabor filters for feature extraction and the nearest neighbor technique for classification have been investigated. The resulting face-recognition design can be used in a door access control system. In this study, a Xilinx system generator and an ISE project navigator were used to design the required simulation and to produce the hardware implementation reports. The distributive arithmetic FIR filter was used in the feature extraction stage to compute the convolution operation between the input image (three region images) and each of the 40 Gabor matrices (filter coefficients). In the classification stage, the simulation design of the nearest neighbor technique was attempted based on the City Block distance. The results obtained using the simulations confirmed the feasibility of implementing a face recognition system on an FPGA device with minimum hardware. The process of converting the MATLAB code of Gabor filters into 40 fixed matrices can reduce the hardware resources to only one 8.192KB RAM for each filter. Since the Xilinx simulation provides a one-to-one realistic correspondence between simulation and real implementation, this validates our research and will soon be implemented on a Xilinx FPGA device. This research will serve as the foundation for future studies and further improvement in both system reliability and face recognition accuracy, leading to the design of a door access control system based on face recognition. The required time processing and the critical path delay of the FPGA device will be discussed in future research when the hardware implementation and camera modeling are completed. Future work will focus on reducing the processing time delay by either reducing the number of the selected Gabor filter orientations and scales or by dividing the resulting Gabor matrix into sub-matrices and applying these sub-matrices to the system in parallel at the same time by increasing the hardware utilization of the FPGA device.

Since we are trying to develop a prototype, we cannot determine the operational scale of our device. But once it is designed, tested, and implemented we would gradually get to know about its scale of information. Based on the cost and accuracy effectiveness, we believe that once our device is implemented, it will be well accepted by the society.

5.2.2 Economic Feasibility

Economic feasibility analysis is the most frequently used method for evaluating the effectiveness of a new or proposed system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with the costs needed to be spending for

implementation and operation. In case, the benefits outweigh costs, the decision is made to design and implement the system.

The determination of economic feasibility requires an identification of the potential costs associated with the implementation of the proposed system. When multiple layers are added to the existing devices, the cost may be increased by 50 percent to 100 percent, which is still a feasible price for a high security authentication device. There are a few issues which may potentially skew the calculations for the economic feasibility study. The first is the advancement of technology in materials and software used for the proposed system which may increase the price accordingly. The second is the reduction in price if the devices are manufactured in bulk.

In terms of cost benefits, the banking and financial organizations are expected to appreciate the benefits immediately. An increase in efficiency, a lower duration of a down time due to increased troubleshooting ability, and a potential for new users are all benefits of the system. The proposed model with thorough cost and risk analysis shows that the system is economically feasible to be implemented as a profitable authentication system with assured security from imposters.

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Our device is a combination of both hardware and software. The hardware components are easily available in the market at a very cheap rate. Also the software used in this project is a free source software.

5.2.3 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Technical feasibility study is carried out to determine whether the proposed system has the capability in terms of software, hardware, personnel, and expertise to handle the completion of the project. The authentication systems using 3D biometrics are tolerable for circumvention. The drawback of 3D physiological and behavioural biometric recognition includes high cost, lack of powerful image processing algorithms,

and low accuracy of acquisition devices. As technology advances in the manufacturing of high speed processors and huge storage disks, the above mentioned problems are easily solved in future. Handwritten 2D signature acquisition devices are already available with assured success in signature analysis. Moreover, the proposed system utilizes the same principle of 2D signature pads with multiple devices placed in a stack and connected to a computer. The handwritten signatures of multiple layers are collected and analyzed simultaneously to produce the result of the authentication process. The proposed model with proven results through the prototype shows that the system is technically feasible to be implemented as an independent authentication system.

The camera system is compact and can be implemented with low cost. The implemented face detection algorithm (Haar like cascade classifier) is very effective, with an accuracy of 88.9 percent which can be increased further by effectively improving the illumination of the area. However, this system is connected with the help of an Ethernet cable to the laptop to communicate with the raspberry pi. This can be overcome by making the system wireless.

5.2.4 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of

Chapter 6

System Design

6.1 Introduction

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.2 System Development Methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.2.1 Model Phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

- Requirement Analysis: This phase is concerned about collection of require- ment of the system. This process involves generating document and requirement review.
- System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm, data structure, software architecture etc.
- Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

- Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation
- Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.
- Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.2.2 Reason for choosing waterfall model as development method

- Clear project objectives.
- Stable project requirements.
- Progress of system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- Logic of software development is clearly understood.
- Production of a formal specification
- Better resource allocation.
- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources required as once one phase is finished those people can start working on to the next phase.

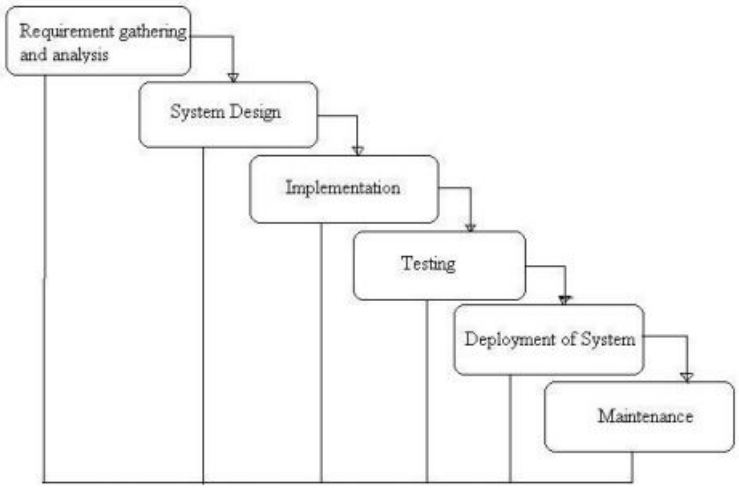

Figure 6.1: Waterfall Model

Line	Symbol
Association	AssociationName
Aggregation	
Generalization	
Dependency	
Activity edge	
Event, transition	event[guard]/action
Link	linkName
Composition	
Realization	
Assembly connection	
Message	some code
Control flow	

Table 6.1: Symbols used in UML

6.3 DESIGN USING UML

Designing UML diagram specifies, how the process within the system communicates along with how the objects with in the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

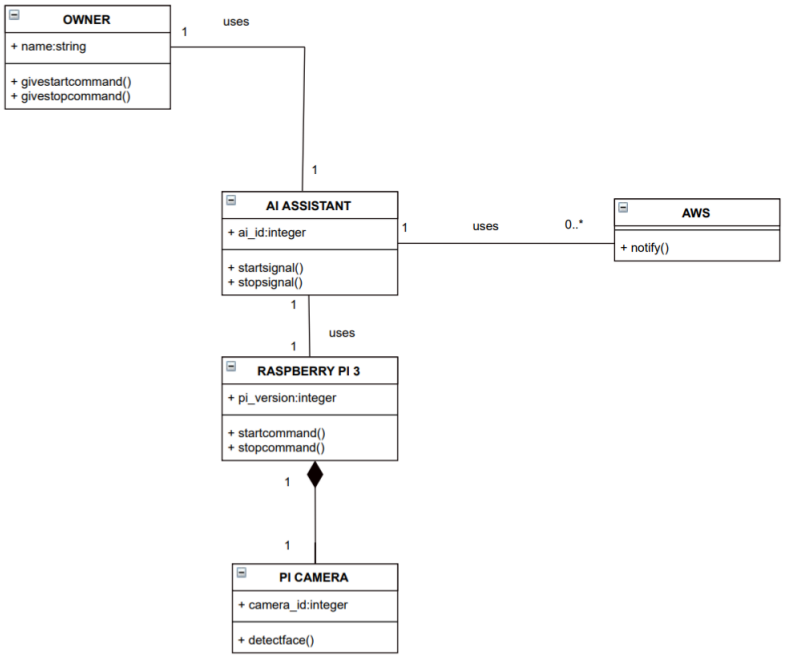

Figure 6.2: Class Diagram

6.4 CLASS DIAGRAM

UML class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

6.5 ACTIVITY DIAGRAM

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition triggered by the completion.

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

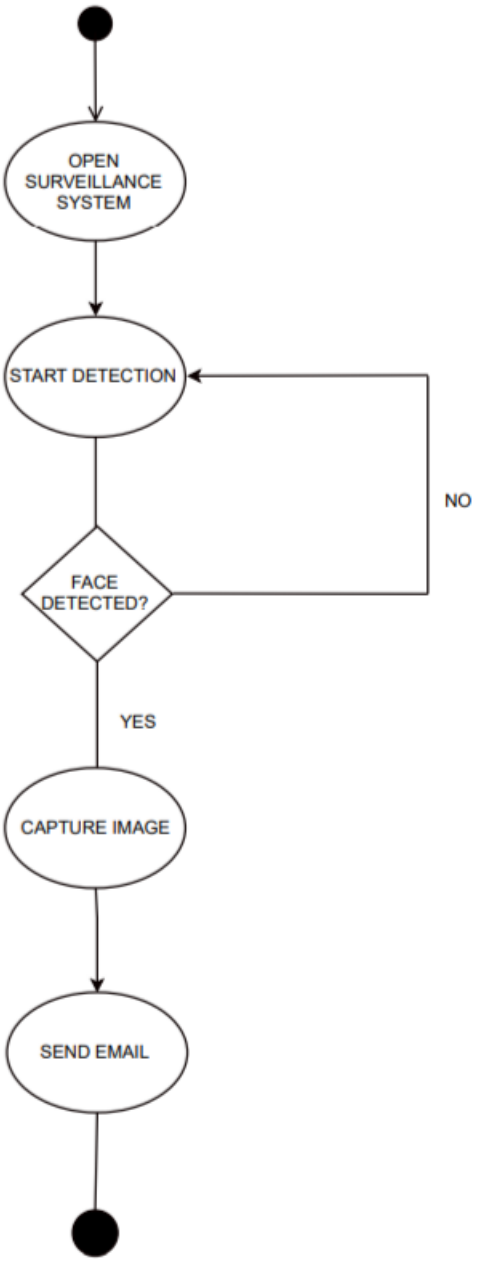

Figure 6.3: Activity Diagram

6.6 SEQUENCE DIAGRAM

Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction. **Basic elements:**

- **Vertical rectangle:** Represent the object is active (method is being performed).
- **Vertical dashed line:** Represent the life of the object.
- **X:** represent the life end of an object. (Being destroyed from memory)
- **Horizontal line with arrows:** Messages from one object to another.

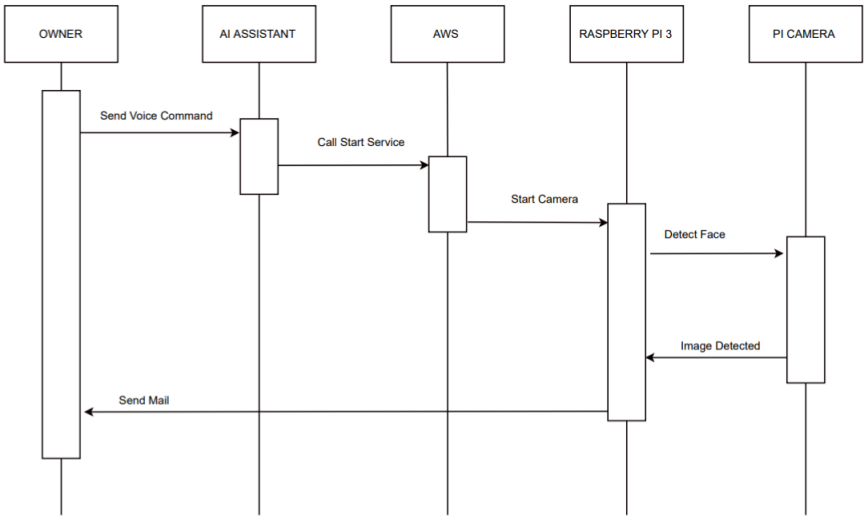

Figure 6.4: Sequence Diagram

6.7 USE CASE DIAGRAM

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

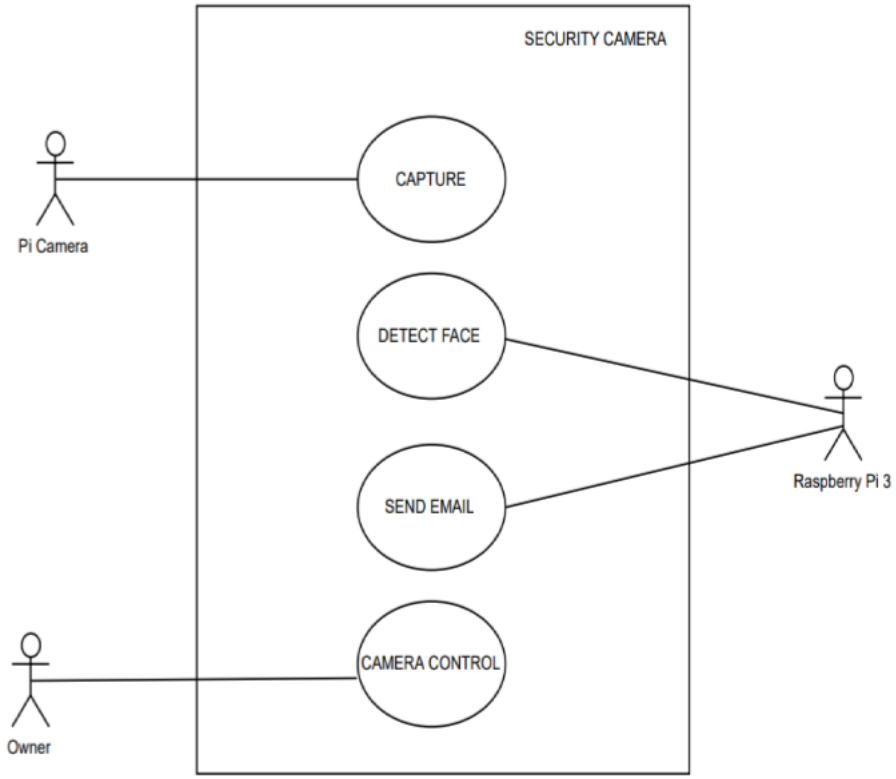

Figure 6.5: Use Case Diagram

Chapter 7

Implementation

7.1 Introduction

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into a best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow. The implementation stage requires the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform.
- Appropriate selection of the language for application development.

7.2 Block Diagram

The block diagram of the paper is quite simple which has a few basic components but it is quite efficient in producing the result as required.

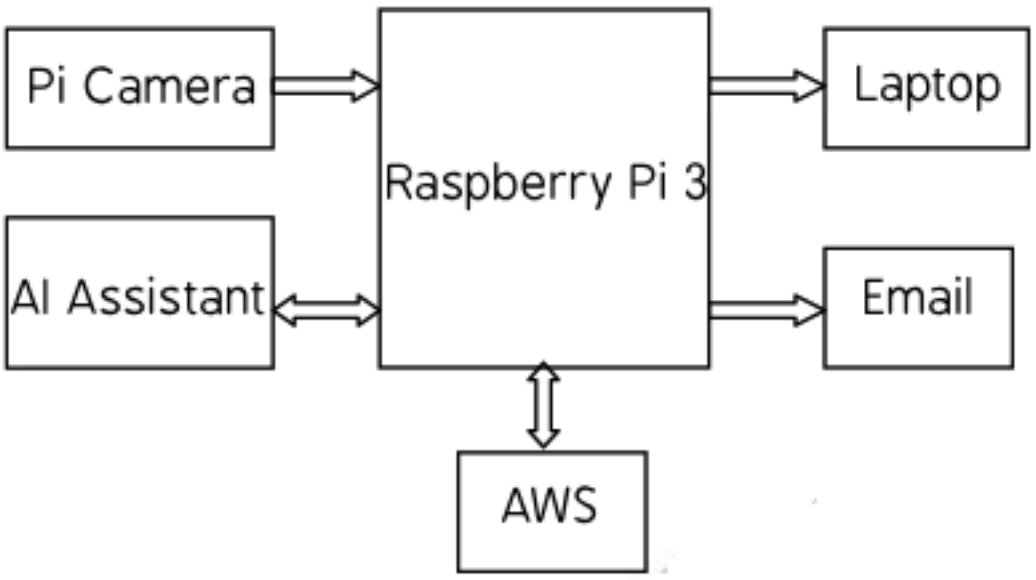

Figure 7.1: Block diagram of the system

The input is the pi camera, which is used to capture the image and the captured image is sent to the processor which checks for the faces. If any faces are detected then it is further processed to check if the face is familiar or not. Finally the output is produced.

7.3 Working Principle

The overall working of the camera can be explained with the help of a flowchart. The image is captured by the picamera which has 5MP pixel resolution with 30 FPS, this image is then sent to the face detection module, which checks the frame obtained for any faces that can be found with the help of the Haar like features, if the face is detected then it is cropped out.

Once the face is compared with the well trained database, it is checked if the face is recognized. If the image matches with the database, then the person is a visitor and a message is sent to the user via a GSM module indicating that someone who is known has come home. However, if the face doesn't match with the database, then the person is identified as a stranger, and SMS is sent alerting the user, and an audio output is produced to warn and alarm the intruder.

7.4 Face Detection

We are providing a secure system, whose in ut I,0captured from the pi camera is sent to processor for face detection. The algorithm used for face detection is Haar like feature cascade classifier.

Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector.[2] Viola and Jones adapted the idea of using Haar wavelets and developed the so-called Haar-like features. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize sub-sections of an image. Therefore a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target face.

The captured image is first converted into the numpy which is a multidimensional array supported by the openCV. Now this image is converted to gray scale, with the help of the loaded haar cascade file from the openCV documentation, the feature is

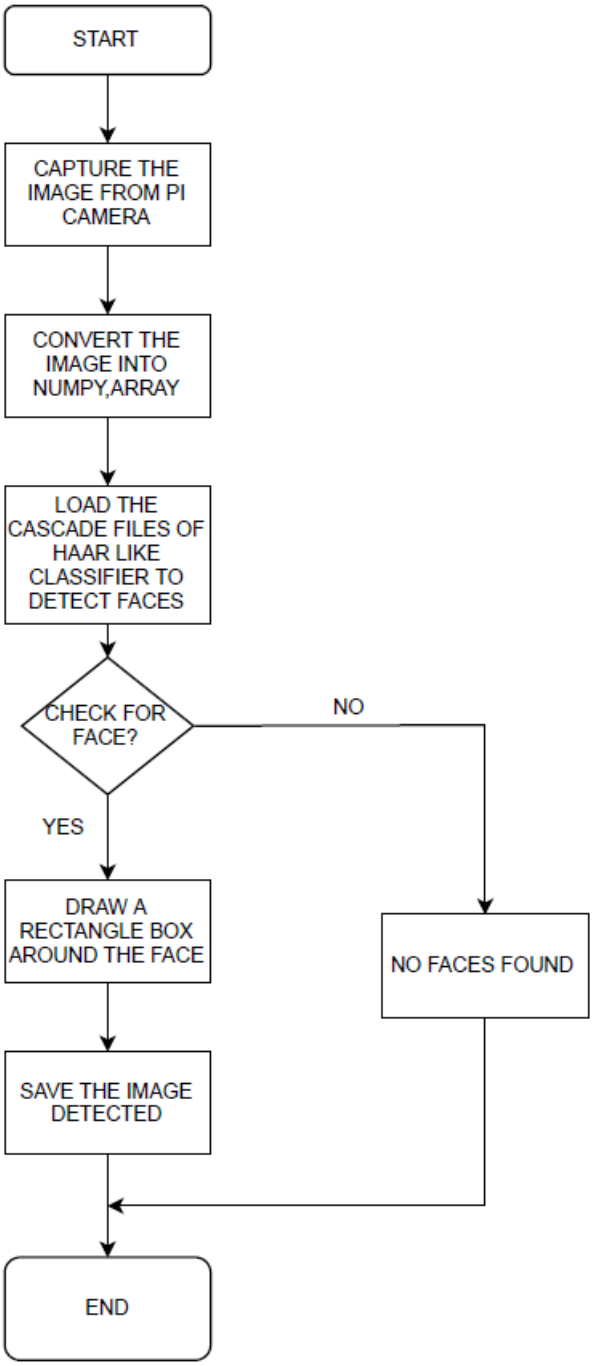

Figure 7.2: Haar Like Feature Cascade Classifier

compared with the image, if any face is found based upon the haar like feature, [3]a rectangle box is drawn to indicatethat a face is detected.

7.5 Code

```python
import smtplib
from email.mime.Multipart import MIMEMultipart
from email.mime.Text import MIMEText
from email.mime.Image import MIMEImage

# Email you want to send the update
fromEmail = '@gmail.com'

fromEmailPassword = ''

# Email you want to send the update to
toEmail = '@gmail.com'

def sendEmail(image):
    msgRoot = MIMEMultipart('related')
    msgRoot['Subject'] = 'Security Update'
    msgRoot['From'] = fromEmail
    msgRoot['To'] = toEmail
    msgRoot.preamble = 'Raspberry pi security camera update'

    msgAlternative = MIMEMultipart('alternative')
    msgRoot.attach(msgAlternative)
    msgText = MIMEText('Smart security cam found object')
    msgAlternative.attach(msgText)

    msgText = MIMEText('<img src="cid:image1">', 'html')
    msgAlternative.attach(msgText)

    msgImage = MIMEImage(image)
    msgImage.add_header('Content-ID', '<image1>')
    msgRoot.attach(msgImage)

    smtp = smtplib.SMTP('smtp.gmail.com', 587)
    smtp.starttls()
    smtp.login(fromEmail, fromEmailPassword)
    smtp.sendmail(fromEmail, toEmail, msgRoot.as_string())
    smtp.quit()
```

Figure 7.3: Sends Mail

```python
import cv2
import sys
from mail import sendEmail
from flask import Flask, render_template, Response
from camera import VideoCamera
import time
import threading

email_update_interval = 600
video_camera = VideoCamera(flip=True)
object_classifier = cv2.CascadeClassifier("models/fullbody_recognition_model.xml")

app = Flask(__name__)
last_epoch = 0

def check_for_objects():
    global last_epoch
    while True:
        try:
            frame, found_obj = video_camera.get_object(object_classifier)
            if found_obj and (time.time() - last_epoch) > email_update_interval:
                last_epoch = time.time()
                print ("Sending email...")
                sendEmail(frame)
                print ("done!")
        except:
            print ("Error sending email: ", sys.exc_info()[0])
@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
@app.route('/video_feed')
def video_feed():
    return Response(gen(video_camera),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
if __name__ == '__main__':
    t = threading.Thread(target=check_for_objects, args=())
    t.daemon = True
    t.start()
    app.run(host='0.0.0.0', debug=False)
```

Figure 7.4: Launch Program

```python
import cv2
from imutils.video.pivideostream import PiVideoStream
import imutils
import time
import numpy as np

class VideoCamera(object):
    def __init__(self, flip = False):
        self.vs = PiVideoStream().start()
        self.flip = flip
        time.sleep(2.0)

    def __del__(self):
        self.vs.stop()

    def flip_if_needed(self, frame):
        if self.flip:
            return np.flip(frame, 0)
        return frame

    def get_frame(self):
        frame = self.flip_if_needed(self.vs.read())
        ret, jpeg = cv2.imencode('.jpg', frame)
        return jpeg.tobytes()

    def get_object(self, classifier):
        found_objects = False
        frame = self.flip_if_needed(self.vs.read()).copy()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        objects = classifier.detectMultiScale(
            gray,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags=cv2.CASCADE_SCALE_IMAGE
        )

        if len(objects) > 0:
            found_objects = True
        # Draw a rectangle around the objects
        for (x, y, w, h) in objects:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        ret, jpeg = cv2.imencode('.jpg', frame)
        return (jpeg.tobytes(), found_objects)
```

Figure 7.5: Detect's Face

```python
import boto3
import os
import time

access_key = "AKIAIFZWHWUW6WNJ7U4A"
access_secret = "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"
region ="us-east-1"
queue_url = "https://sqs.us-east-1.amazonaws.com/227679985855/CameraQueue"

def pop_message(client, url):
    response = client.receive_message(QueueUrl = url, MaxNumberOfMessages = 10)

    #last message posted becomes messages
    message = response['Messages'][0]['Body']
    receipt = response['Messages'][0]['ReceiptHandle']
    client.delete_message(QueueUrl = url, ReceiptHandle = receipt)
    return message

client = boto3.client('sqs', aws_access_key_id = access_key,
    aws_secret_access_key = access_secret, region_name = region)

waittime = 20
client.set_queue_attributes(QueueUrl = queue_url, Attributes =
    {'ReceiveMessageWaitTimeSeconds': str(waittime)})

time_start = time.time()
while (time.time() - time_start < 60):
        print("Checking...")
        try:
                message = pop_message(client, queue_url)
                print(message)
                if message == "on":
                        #os.system("~/cameraon.sh")
                        print("Camera On")
                elif message == "off":
                        #os.system("~/cameraoff.sh")
                        print("Camera Off")
        except:
                pass
```

Figure 7.6: Connects to AWS Services

Chapter 8

Testing

8.1 Introduction

Testing is an important phase in the development life cycle of the product this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Once the implementation is done, a test plan should be developed and run on a given set of test data. Each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is suppose to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as the solution to theoriginal problem.
- To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code. The test cases executed for this project are listed below. Description of the test case, steps to be followed; expected result, status and screenshots are explained with each of the test cases.

8.2 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important types of testing are:

8.2.1 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.

- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity.

8.2.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works. It uncovers a different class of errors in the following categories:

- Incorrect or missing function.
- Interface error.
- Performance errors.
- Initialization and termination errors
- Error in objects.

Advantages:

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

8.2.3 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

8.3 Test Case 1

Function: def sendmail(image)
Purpose: Send mail to the user
Preconditions: The object should be detected first
Inputs: Face detected
Expected Output: Send mail to the user
Postconditions: Mail sent successfully

8.4 Test Case 2

Function: def checkforobject()
Purpose: Face detection
Preconditions: Camera should be turned on
Inputs: Image Feed
Expected Output: Detect Image
Postconditions: Image Detected successful

8.5 Results

The smart surveillance camera is very effective in a way that it provides security by reducing the alarming raise of crime at home. The Face of the human being is detected easily with the help of the implemented algorithm a Haar classifier. like cascade.

Chapter 9

Conclusion And Future Scope

9.1 Conclusion

Thus, we have developed a smart surveillance camera that can be started using alexa and is capable of providing face detection a. Also the camera system is compact and can be implemented with low cost. The implemented face detection algorithm (Haar like cascade classifier) is very effective, with an accuracy of 88.9 percent which can be increased further by effectively improving the illumination of the area. However, this system is connected with the help of a Ethernet cable to the laptop to communicate with the raspberry pi. This can be overcome by making the system wireless.

9.2 Future Scope

This system has a wide range of uses in various fields, such as banking, forensic department, etc The reason this system is quiet useful is due to the fact that it is highly compact and it provides face detection and an instant notification about the same through email. In addition to this face recognition can also be tried in future. Recognition is the main part of any security system. Usually for a best recognition system, we require a well-trained database, which can provide the base for our recognition. So to obtain the database, first collect the images of the subject individual for the recognition. Once we obtain and train our system, we can provide face recognition.

We use the local binary pattern histogram (LBPH) for providing face recognition. This method helps us to provide a recognition model. The image is converted into a gray scale image. Then, the image pixels are compared with the neighboring pixels in a clock-wise or anti-clock-wise manner. Histogram is performed and normalization is done and a feature vector is generated for every image. These feature vectors can now be processed with some algorithms to classify images which is used to identify the texture. Once the face is recognized, it is checked to see if the detected face is familiar or not. Thus we integrate the face detection and recognition to provide a smart surveillance system for the domestic purposes in our everyday life.

References

[1] Wilson Feipeng Abaya, a Low cost smart security camera with night vision capability, De La Salle Univ., Manila, Philippines

[2] Andrew S. Tanenbaum:*Operating Systems Design and Implementation*, Prentice Hall, 2006

[3] About IPTV on Wikipedia `http://en.wikipedia.org/wiki/IPTV`

[4] About VNC on Wikipedia `http://en.wikipedia.org/wiki/Virtual_Network_Computing`

[5] LibVNC server `http://libvncserver.sourceforge.net`

[6] DirectFB documentation `http://elinux.org/DirectFB`

[7] jointSPACE documentation `http://sourceforge.net/apps/mediawiki/jointspace/index.php?title=Main_Page`

[8] PuTTy on Wikipedia `http://en.wikipedia.org/wiki/PuTTy`

[9] Nicola L. C Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. *In Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.