**HOME SECURITY SYSTEM**

A PROJECT REPORT

Submitted by:

**ATHIRA A S**

**LMC16MCA2**

*to*

*The APJ Abdul Kalam Technological University*

*in partial fulfillment of the requirements for the award of the Degree*

*of*

*Master of Computer Applications*

**Department o**

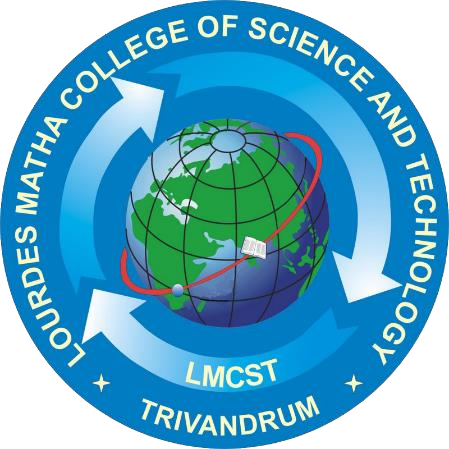
**f**

**Computer Applications**

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

KUTTICHAL, THIRUVANANTHAPURAM 695574

MAY 2019



# HOME SECURITY SYSTEM

A PROJECT REPORT

Submitted by:

**ATHIRA A S**

## LMC16MCA2

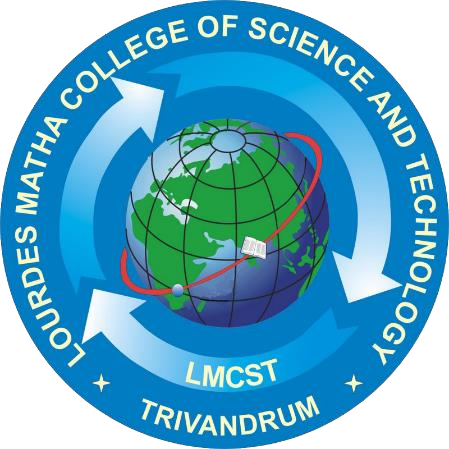
*to*

*The APJ Abdul Kalam Technological University*

*in partial fulfillment of the requirements for the award of the Degree*

*of*

*Master of Computer Applications*



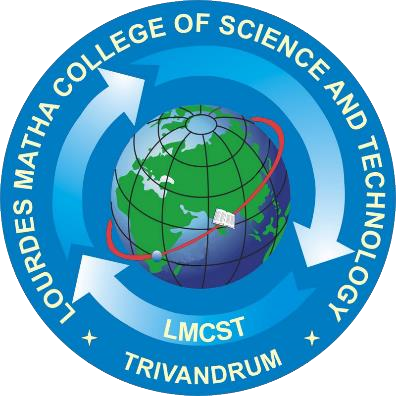
## Department Of Computer Applications

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

KUTTICHAL, THIRUVANANTHAPURAM 695574

MAY 2019

## DEPARTMENT OF COMPUTER APPLICATIONS LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY KUTTICHAL, THIRUVANANTHAPURAM



## CERTIFICATE

This is to certify that the report entitled ‘**Home Security System’** submitted by **Athira A**

**S** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision.

Prof. Justin G Russel (Internal Supervisor)

Prof. Neethu Mohan Prof. Selma Joseph

(Project Co-ordinator) (Head of the Dept.)

## DECLARATION

I undersigned hereby declare that the project report ‘Home Security System’, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof.Justin G Russel. This submission represents my ideas in my own words and, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University.

Place: Trivandrum Signature

Date: 30/05/2019Athira A S

|  |  |  |  |
| --- | --- | --- | --- |
| CONTENTS |  |  |  |
| ACKNOWLEDGEMENT |  |  | i |
| ABSTRACT |  |  | ii |
| Chapter 1. Introduction |  |  |  |
| 1.1 General Background |  |  | 01 |
| 1.2 Objective |  |  | 01 |
| Chapter 2. Literature Survey |  |  |  |
| 2.1 Study of similar work |  |  | 02 |
| 2.1.1 Existing System |  |  | 03 |
| 2.1.2 Drawbacks of Existing System    Chapter 3. Overall Description |  |  | 03 |
| 3.1 Proposed System |  |  | 04 |
| 3.2 Features of Proposed System |  |  | 04 |
| 3.3 Functions of Proposed System |  |  | 05 |
| 3.4 Requirements Specification |  |  | 05 |
| 3.5 Feasibility Analysis |  |  | 06 |
| 3.5.1 Technical Feasibility |  |  | 07 |
| 3.5.2 Operational Feasibility |  |  | 07 |
| 3.5.3 Economical Feasibility |  |  | 08 |
| 3.5.4 Behavioural Feasibility    Chapter 4. Operating Environment |  |  | 08 |
| 4.1 Hardware Requirements |  |  | 09 |
| 4.2 Software Requirements |  |  | 09 |
| 4.3 Tools and Platforms |  |  | 10 |
| 4.3.1 Android Language |  |  | 10 |
| 4.3.2 Python Language |  |  | 10 |
| 4.3.3 Windows 7 |  |  | 11 |
| 4.3.4 Raspberry pi |  |  | 11 |

Chapter 5. Design

5.1 System Design 12

5.1.1 Data Flow Diagram 13

5.1.2 Project DFD 15

5.2 Database Design 16

5.3 Input Output Design 17

5.4 Program Design 18

Chapter 6. Functional and Non-Functional Requirements

6.1 Functional requirements 20

6.1.1 Image based evaluation on face recognition 20

6.1.2 Owner based application 20

6.2 Non-Functional Requirements 21

Chapter 7. Testing

7.1 Testing Strategies 22

7.2 Unit Testing 23

7.3 Integration Testing 24

7.4 User Acceptance Testing 24

7.5 Data Validation Testing 24

7.6 White Box Testing 25

7.7 Black Box Testing 25

7.8 Output Testing 25

7.9 Testing Result 26

Chapter 8. Results and Discussion

8.1 Results 28

8.2 Screen Shots 28

Chapter 9. Conclusion

9.1 System Implementation 31

9.2 Conclusion 31

9.3 Future Enhancement 32

Bibliography I

* + 1. Books
    2. Website

Appendices II

* + 1. SCRUM Board
    2. List of Tables
    3. List of Figures
    4. Abbreviations and Notation
    5. Coding

# ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role of expressing my gratitude.

First of all I would like to thank God almighty for bestowing us with wisdom, courage and perseverance which had helped us to complete this project **Home Security System**. This project has been a reality as a result of the help given by a large number of personalities.

I am extremely thankful to **Rev.Dr. Tomy Joseph Padinjareveettil,** Director and **Prof. Mohanlal P P** , Principal, Lourdes Matha College of Science And Technology Kuttichal for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I would like to remember with gratitude **Prof. Selma Joseph**, Head Of Department, Department of Computer Applications, Lourdes Matha College of Science And Technology Kuttichal, for the encouragement and guidance rendered.

I express our profound and sincere gratitude to **Prof. Neethu Mohan**, Department of Computer Applications, Lourdes Matha College of Science And Technology Kuttichal for the immense support and guidance provided.

I express my sincere thanks to **Prof. Justin G Russel**, Department of Computer Applications, Lourdes Matha College of Science And Technology Kuttichal for his valuable guidance, support and advices that aided in the successful completion of my project.

Finally, I wish to express my sincere gratitude to all our friends, who directly or indirectly contributed in this venture.

i

# ABSTRACT

The system is about remotely managed door accessibility and voice altering through smart phone and receive captured image of visitor at door bell system has indispensable in daily life. The design and development of a smart security system, based on face recognition, pre-image processing human motion detection and remotely monitoring technology.

This project describes about the Implementation and deployment of wireless control system and accessibility into a home environment. A camera module is used to detect face recognition on the basis of deep learning. Electromagnetic door lock module operate the door accessibility.

The proposed system uses controller interface system with Raspberry Pi which is low cost and consume smaller amount of power. When visitor motion detected at Door, Camera module interfaced to Raspberry Pi capture images, save it on system. When a visitor is at the door the user is notify via an app and user can authorized the visitor entry in the house. The concerned authority can control the system and view video stream of camera module through Smart mobile Phone. The system also provided concerned authority to use Smart Phone to send command for voice alert when intruder identified.

ii

**CHAPTER 1**

# INTRODUCTION

## 1.1 GENERAL BACKGROUND

This project is a significant step towards smart home and living. With the increase in trend of online shopping combined with conventional trends of delivery of official documents through parcels, couriers and so on we realized that a major inconvenience is faced by the customer as well as the delivery person if the concerned one is not present at his home at time of delivery.

This also extends to friends and relatives who may visit your place unannounced.

## 1.2 OBJECTIVE

There is no smart means through which the owner of the house is notified about the visitor in case he is outdoors or unable to hear the bell. Using the microcomputer, ‘Raspberry pi’, the smart doorbell, hence solves the problem of visitors remaining unattended in case the concerned person is not available. The home security system helps you to know the visitors who visits your home by sending the image through the android application.

**CHAPTER 2**

# LITERATURE SURVEY

## 2.1 STUDY OF SIMILAR WORKS

### Wired Doorbell

In most wired systems, a button on the outside next to the door, located around the height of the doorknob. Activates a signaling device (usually a chime, bell, or buzzer) inside the building. Pressing the doorbell button, a single-pole, single throw (SP ST) pushbutton switch momentarily closes the doorbell circuit. One terminal of this button is wired to a terminal on a transformer. A doorbell transformer steps down the 120 or 240-volt AC electrical power to a lower voltage, typically 10 to 20 volts. The transformer's other terminal connects to one of three terminals on the signaling device. Another terminal is connected to a wire that travels to the other terminal on the button. Some signaling devices have a third terminal, which produces a different sound. If there is another doorbell button (typically near a back door), it is connected between the transformer and the third terminal. The transformer primary winding, being energized continuously, does consume a small amount (about 1 to 2 W) of standby power constantly; systems with lighted pushbutton switches may consume a similar amount of power per switch. The tradeoff is that the wiring to the button carries only safe, low voltage isolated from earth ground. A common signaling device is a chime unit consisting of two flat metal bar resonators, which are struck by plungers operated by two solenoids. The flat bars are tuned to two pleasing notes. When the doorbell button is pressed, the first solenoid's plunger strikes one bar, and when the button is released, a spring on the plunger pushes the plunger up, causing it to strike the other bar, creating a two-tone sound ("ding-dong"). If a second doorbell button is used, it is wired to the other solenoid, which strikes only one of the bars, to create a single-tone ("ding") sound.

### Wireless Doorbells

In recent decades, wireless doorbell systems that do not require wall wiring have become popular. The doorbell button contains a built-in radio transmitter powered by a battery. When the button is pushed, the transmitter sends a radio signal to the receiver unit, which is plugged into a wall outlet inside the building. When the radio signal is detected by the receiver, it activates a sound chip that plays the sound of gongs through a loudspeaker either a two-note ""ding-dong" sound or a longer chime sequence such as Westminster Quarters. To avoid interference by nearby wireless doorbells on the same radio frequency, the units can usually be set by the owner to different radio channels.

#### 2.1.1 Existing System

The existing doorbell systems follow traditional approach. When a visitor presses the switch be that rings inside the house. If someone is present in the house they open the door and if no one is present in the house the visitor waits for certain time and leaves the place without any clue. Doorbells have been playing an important role in protecting the security of modern homes since they were invented. A doorbell allows visitors to announce their presence and request entry into a building as well as enables the occupant to verify the identity of the guests to help prevent home robbery or invasion at a moment's notice. There are two types of doorbells depending on the requirement of wall wiring.

#### 2.1.2 Drawbacks Of Existing System

* Lack of safe and security
* Owner cannot identify who visited and leave the home
* Less efficient

**CHAPTER 3**

# OVERALL DESCRIPTION

## 3.1 PROPOSED SYSTEM

Our proposed system will be helpful for those who aren’t at home most of the times and need to keep track on visitor. The proposed system uses controller interface system with Raspberry Pi which is low cost and consume smaller amount of power. When visitor motion detected at Door, Camera module interfaced to Raspberry Pi capture images, save it on system. When a visitor is at the door the user is notify via an app and user can authorized the visitor entry in the house .The concerned authority can control the system and view video stream of camera module through Smart mobile Phone. The system also provided concerned authority to use Smart Phone to send command for voice alert when intruder identified. Users can monitor visitors and control the door lock on active SSH (Secure Shell) page designed on android platform and enhanced with JavaScript. This system finds a wide application in areas where physical presence is not possible all the time.

## 3.2 FEATURES OF PROPOSED SYSTEM

* Efficient
* Cost effective
* User friendly
* Time saving

### 3.3 FUNCTIONS OF PROPOSED SYSTEM

* Security: The doorbells with a camera attached to the panel on outside door helps a lot in enhancing the safety of our house.
* Human effort can be reduced.
* Portability : The wireless doorbell can be kept at different places according to the needs
* The developed system will capture the visitor’s image and compares with the registered image from the database using deep learning. And send the image to the user’s android app.
* Cost effective

## 3.4 REQUIREMENT SPECIFICATION

System analyst talk to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaire’, detailed investigation, observation, collection of samples etc. As the details are collected, the analyst study the requirements data to identify features the new system must have, including both the information the system should produce and operational features such as processing controls, response times and input-output methods.

Requirements specification simply means, “Figuring out what is to be made before making it.” It determines what people need before starting to develop a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements. These should reflect what consumer wants.

The requirements of HOME SECURITY SYSTEM are: Raspberrry pi, Camera, Mobile phone Opencv, Windows 7 or higher, Python, Android, Android studio, Tiny DB

## 3.5 FEASIBILITY ANALYSIS

The prime objective of feasibility study is to ensure that the problem is worth to be solved. At the stage a cost benefit is performed to assertion that the benefit from the system will over rule association with the whole analysis, design and development of the new system. As important outcome of the preliminary investigation determining whether the system required is feasible.

Steps in Feasibility Analysis

Feasibility Analysis involves eight steps:

* Form a project team and appoint a project leader.
* Prepare a system flow chart
* Enumerate potential candidate system.
* Describe and identify characteristics of candidate systems.
* Describe and evaluate performance and cost effectiveness of each candidate systems.
* Weight system performance and cost data.
* Select the best candidate system.
* Prepare and report final project directive and management.

The proposed system is tested in all three aspects of feasibility.

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility
* Behavioral Feasibility

### 3.5.1 Technical Feasibility

The main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. Investing the existing system in the area under investigation and generating ideas about the new system does this. Feasibility study has been done to gather required information. Data was gathered and checked for completeness and accuracy. Analyzing the data involved identification of the components of the system and their interrelationship and identified the strength and weakness of the system.

Our system is developed by using front end as Python & Android and back end as TinyDB. It is technically feasible and it has lot of features. Python & Android are secure languages. So the technical part of this project is very secure. The system that developed by the combination of these two languages has high performance. It is highly reliable, so our system is highly reliable.

Maintainability and productivity area also high. So our system is technically feasible.

### 3.5.2 Operational Feasibility

There is no difficulty in implementing the system. The proposed system is effective, user friendly and functionally. The user of the system must be completely unaware of the internal working of the system so that the users will not face any problems running the system. In our system we use Raspberry Pi it act as a stronger Linux OS and we can extend or add any features by using Raspberry Pi.

Design is the only ways that can accurately translate customer registration into finished software or system. Without software design, the risk of building an unstable system exists. System design provides the procedural details necessary for implementing the system recommended in the feasibility study.

### 3.5.3 Economic Feasibility

Economic and Financial analysis is used for evaluating the effectiveness of the system. The project is technically and operationally feasible.

Python & Android is an open source, cross platform. So no cost needs to buy the backend. My system is economically feasible because the project completed in few months. So fewer resources are used and the cost hardware resources is affordable. We can add general features or new features by the use of Raspberry Pi with minimum cost. Excepting cost for developing the system is Rs 7000/-

**3.5.4 Behavioral Feasibility**

The behavioral feasibility depends upon whether the system performed in the expected way or not. Feasibility study is a test of system proposal according to it workability, impact on organization, ability to meet user’s need and effective use of resources. However, a feasibility study provides a useful starting point for full analysis.

Our system is behaviorally feasible because of the effective use of the resources and also the

system satisfied the user needs and the system is user friendly.

# CHAPTER 4

**OPERATING ENVIRONMENT**

## 4.1 HARDWARE REQUIREMENTS

Processor : Intel Pentium4, Raspberry Pi

RAM : 1GB or above

Hard Disk : 250GB or above

Display Size : 14” Color Monitor or above

Screen Resolution : 1024× 768 pixels or above

|  |  |  |
| --- | --- | --- |
| Keyboard |  | : PC/AT Enhanced Type |
| Mouse |  | : Logitech PS/2 Port Mouse |
| Modem |  | : 56Kbps or above |
| Camera |  | : 8MP or above |

## 4.2 SOFTWARE REQUIREMENT

|  |  |  |
| --- | --- | --- |
| Operating System | : | Raspbian os, Windows 7 or higher |
| Front End | : | Python, Android |
| Back End | : | Tiny DB, Mongo DB |
| Server | : | Flask |
| IDE | : | Android studio |

### 4.3 TOOLS AND PLATFORMS

#### 4.3.1 Android Language

Android is an open source and Linux-based Operating System for mobile devices such as smart phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007, whereas the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance. The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

#### 4.3.2 Python Language

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.

#### 4.3.3 Windows 7

An operating system introduced in 2001 from Microsoft’s Windows family of operating systems, the previous version of Windows being *Windows Me.* Microsoft called the release its important product since Windows 95. Along with a redesigned look and feel to the user interface, the new operating system is built on the Windows 2000 kernel, giving the user a more stable and reliable environment than previous versions of Windows.

Windows 7 is a version of the Windows NT operating system. Windows 7 was released to manufacturing on July 22. 2009 and became generally available on October 22, 2009, less than three years after the release of its predecessor, Windows Vista. Window 7’s server counterpart, Windows Server 2008 R2, was released at the same time. Windows 10 is the latest release of Microsoft’s Windows operating system and the successor to the underwhelming Windows 8.

#### 4.3.4 Raspberry Pi

The Raspberry Pi, a tiny single-board computer, has revolutionized the way in which computer science is being taught in schools. It has also turned out to be a boon for software developers. Currently, it has gained popularity much beyond its target market and is being used in robotics projects. Raspberry Pi, a small development board minicomputer that runs the Linux operating system, was developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools in the UK and in developing countries. Raspberry Pi has USB sockets, which support various peripheral plug-and-play devices like the keyboard, the mouse, the printer, etc. It contains ports like HDMI (High Definition Multimedia Interface) to provide users with video output. Its credit-card-like size makes it extremely portable and affordable. It requires just a 5V micro-USB power supply, similar to the one used to charge a mobile phone.

**CHAPTER 5**

# DESIGN

## 5.1 SYSTEM DESIGN

System design is a reduction of an entire system by studying the various operations performed and their relationships within the system and the requirements of its success. One aspect of design is defining the boundaries of the system and determining whether or not the candidate system should consider other related system.

System can be defined, as an orderly grouping of interdependent components can be simple or complex. The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed .It refers to the technical specifications that will be applied in implementing the candidate system .It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented .In the second step, input data and master files are to be designed to meet requirement of the proposed output. The processing phase’s system’s objectives and complete documentation.

System design has two phases:

* Logical
* Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through .We have to deal with how to take entries required and whether and how to process the user data. Also we have to deal with how to present the data in an informative and appealing format .This design also involves the methodology to store, modify and retrieve data from the data base as per the requirement.

Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software. We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner.

### 5.1.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the” flow” of data through an information system. A data flow diagram can also be used for the visualization of data processing(Structural design).The DFD showing the top level of the system is called “Context Diagram”. It should be overview including basic inputs, processes and outputs. Then it is exploded in to more detailed lower level diagram that shows additional features of the system.

The purpose of DFD is to provide a semantic bridge between users and system developers. The diagrams are graphical, eliminating thousands of words, logical representations, modelling what system does; hierarchical, showing system at any level of details; and Jargon less, allowing user interaction and reviewing.

The goal of data flow diagramming is to have a commonly understood model of a system. The diagram is the basis of structured system analysis. The Data flow diagram, also known as “Bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become program in system design. The bubble represents the data transformations and the lines represent data flows in the system.

### (i) Basic DFD Symbols

* **Rectangles** - representing external entities, which are sources or destinations of data.
* **Arrows** - representing the data flows, this can either be electronic data or physical items. It shows the directional movement of data to and from External Entities, the process and Data Stores.
* **Open-ended rectangles or two parallel lines** – representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.
* **Circle or a Rounded Rectangle**- representing processes, which take data as input, do something to it, and output it. It is used to represent functions.

### (ii) Components of Data Flow Diagram

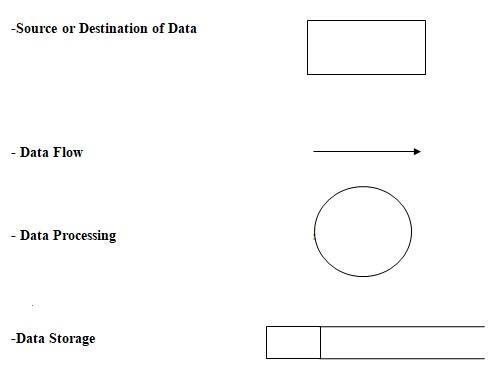


Figure 5.1 DFD Components

#### 5.1.2 PROJECT DFD



Figure 5.2 Context Diagram (Level-0)

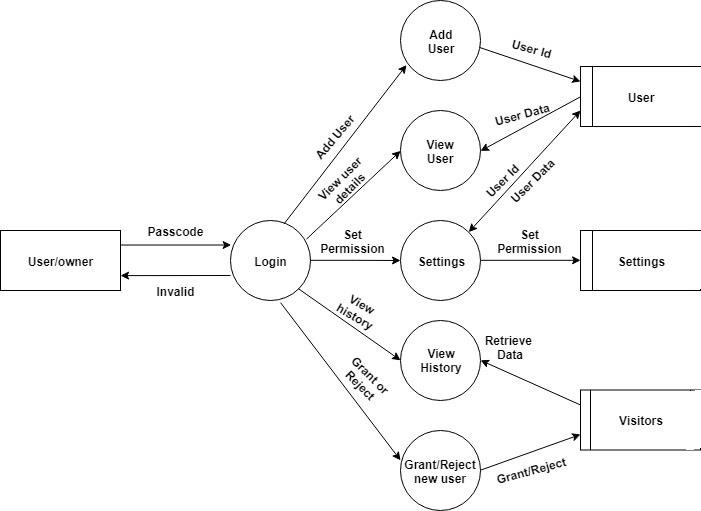


Figure 5.3 Android Application (Level-1)

## 5.2 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve users more quickly and efficiently. The general objective of a database is to make information access easy, quick, inexpensive, integrated and shared by different applications and users. Database design is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for Database Administrators, developers, and other members of joint application development team. The process of creating a data model helps the team uncover additional questions to ask of end users. Effective database design also allows the team to develop applications that perform well from the beginning. By building quality into the project, the team reduces the overall time it takes to complete the project, which in turn reduces project development costs. The central theme behind database design is to "measure twice, cut once". Effective database designers will keep in mind the principles of normalization while they design a database.

TinyDB is a lightweight document oriented database. It's written in pure Python and has no external dependencies. The target are small apps that would be blown away by a SQL-DB or an external database server.

**FEATURES:**

* **Tiny:** The current source code has 1200 lines of code (with about 40% documentation) and 1000 lines tests. For comparison: Buzhug has about 2500 lines of code (w/o tests), CodernityDB has about 7000 lines of code (w/o tests).
* **Document oriented:** Like MongoDB, you can store any document (represented as dict) in TinyDB.
* **Optimized for your happiness:** TinyDB is designed to be simple and fun to use by providing a simple and clean API.
* **Written in pure Python:** TinyDB neither needs an external server (as e.g. PyMongo) nor any dependencies from PyPI.
* **Works on Python 2.7 and 3.3 – 3.6 and PyPy:** TinyDB works on all modern versions of Python and PyPy.
* **Powerfully extensible:** You can easily extend TinyDB by writing new storages or modify the behaviour of storages with Middlewares.
* **100% test coverage:** No explanation needed

## 5.3 INPUT OUTPUT DESIGN

Input designing is the basic theory to be considered during system study. The input media used in the system is the keyboard. Details are entered in the system through different data entry screens. The system is designed in a user-friendly manner. Appropriate error messages are displayed when a false data is entered. Design of the system is web-oriented and is highly interactive to the users. The user interface design is very important for any application. The interface design defines how the software communicates within itself, to system that interpreted with it and with human who use it. The interface design is very good. The user will fall into an interactive software application.

Computer output is the most important one to the user. A major form of the output is the display of the information gathered by the system and the servicing the user requests to the system. Output generally refers to the results or information that is generated by the system. It can be in the form of operational documents and reports. Since some of the users of the system may not operate the system, but merely use the output from the system to aid them in decisionmaking, much importance is given to the output design. Output generation hence serves two main purposes, providing proper communication of information to the users and providing data in a form suited for permanent storage to be used later on. The output design phase consists of two stages, output definition and output specification. Output definition takes into account the type of outputs, its contents, formats, its frequency and its volume. The output specification describes each type of output in detail.

|  |  |  |
| --- | --- | --- |
| **PROCESS** | **INPUT DESIGN** | **OUTPUT DESIGN** |
| Capture photo | Sense for human motion | Face captured Successfully and store it in the database |
| Face recognition | Analyze the captured image | Check whether it is similar |
| Add user | Register with users photo | New user successfully added |
| View user | List of users | View list of users |
| Access granted or deny | Make the users accessibility  Public or private | If it is public then unlock the door .If private lock the door |
| History of visitors | List of visitors | View the list of visitors |

Table: 5.3 input output design

## 5.4 PROGRAM DESIGN

The project is divided into two major modules according to their functionalities.

* Image based evaluation on face recognition
* Owner based application

Face recognition Module

* Step 1: Camera captures the image when any motion detected in front of camera.
* Step 2: Check whether the captured image is human face or not using some algorithms.
* Step 3: Store the captured image in the database
* Step 4: Generate binary tree of the image
* Step 5: When new image come it compare the image with the binary tree of image
* Step 6: if match occurred the face is recognized successfully.
* Step 7: Else face recognition not successful
* Step 8: Stop

Owner Based Application Module

* Step 1: Owner add the new user
* Step 2: Set the accessibility of user as private or public.
* Step 3: If public then the door is unlocked.
* Step 4: If private then the door is locked
* Step 5: Owner can view list of users, list of visitors and can verify and grant permission for visitors/users according to their modes.
* Step 6: Stop

**CHAPTER 6**

# FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

## 6.1 FUNCTIONAL REQUIREMENTS

Functional requirements represent the intended behavior of the system. This behavior may be expressed as services, tasks or functions that the specified system is required to perform.

The system is should be designed to capture all human faces and recognize it through deep learning. The following functional requirements have been identified for this project. The proposed system consists of 2 modules. They are given below:

6.1.1 Image Based Evaluation On Face Recognition

This module uses opencv and python for face recognition. The user needs to show his face to the camera. The camera captures the images of the visitor and store it in the database for face recognition. The system will use opencv functions for identify the corresponding similar faces using the algorithms local binary pattern and haarcascades.

6.1.2 Owner Based Application

In order to make the proposed system user friendly, an android application is used to identify visitors. The face recognized by the system and the corresponding image captured is shown in the owners android application. Also owner can verify and grant permission through his android application.

## 6.2 NON FUNCTIONAL REQUIREMENTS

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. Non-functional requirements are

“system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”,

“quality goals", "quality of service requirements" and "non-behavioural requirements. Some of the non-functional requirements are mentioned below

* **Usability**: The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warnings or alerts shall be specified in clear way.
* **Efficiency**: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way.
* **Interoperability**: The system shall be able to interact with other systems. The system should able to be supported at least one software which has a relationship with payment process
* **Portability**: The system shall be independent of the specific technological platform used to implement it.
* **Reliability**: Reliability defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failure for a particular period of time
* **Availability**: Availability measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365days per year.

**CHAPTER 7**

# TESTING

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences.

Testing is a purpose of executing a programmed with intend of finding errors.

1. Preparing a test case that has high probability of finding undiscovered errors.
2. Testing to erase out all kinds of bucks from the program.

Before going for testing, first we have to decide the type of test. For this impact system, unit testing is carried out. And the following things are taken to consideration.

1. To ensure that information properly places in and out of the program.
2. To ensure that the module operates properly at boundaries established to limit or restrict processing.
3. To find out whether all statements in module have been executed at least once.
4. To find out whether error handling paths are working correctly.

## 7.1 TESTING STRATEGIES

A strategy for software testing integrates software test case design methods in to a well-planned series of steps that results in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and undertaken, and how much effort, time and resources will be required. Therefore any testing strategy must incorporate test planning, test case, design, test execution and resultant data collection and evaluation. A software testing strategy should be flexible enough to promote customized testing approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project processes. The project manager, software engineer and testing specialists develop a strategy for software testing. The general characteristics of software testing strategy are:

1. Testing begins at the component level and works “outward” toward the integration of the entire computer system.
2. Different testing techniques are appropriate at different point in time.

A strategy for software testing must accommodate low-level testis that are necessary to verify a small source code segment has been correctly implemented as well as high level testing that validate major system function against customer requirements.

## 7.2 UNIT TESTING

Unit test comprises of a set test performed by an individual programmer prior to the integration of the unit into large system. Program unit is usually small enough that the programmers who developed and can it in great detail and certainly in greater than will possible when the unit is integrated into evolving software project. Unit testing should be an exhaustive as possible. In this system, each module was tested individually to ensure that every representation in the module meets the requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Procedures | Expected Result | Actual Result | Pass/Fail |
| 1 | Take photo | Photo is captured | Same as expected result | Pass |
| 2 | Training the model | A fair Accuracy | Same as expected result | Pass |
| 3 | Captured photo is send to the owner’s android application | Photo send successfully | Same as expected result | Pass |

Table 7.2 : Unit test cases and results

## 7.3 INTEGRATION TESTING

Integration testing is a system technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. The objective is to take unit testing modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole.

Bottom-up integration consists of a unit test followed by testing of the entire system. Subsystem consists of several modules that communicated with other defined interface.

The errors were isolated and corrected to produce a fully functional system. Top-down integration method is an incremental approach to the construction of the program structure. The project was tested to ensure that every representation meets the requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Procedures | Expected result | Actual result | Pass/Fail |
| 1 | Face recognition | Face recognized successfully | Same as expected | Pass |
| 2 | Establish the accessibility of door | Access grant or deny | Same as expected | Pass |

Table 7.3: Integration test cases and results

## 7.4 USER ACCEPTANCE TESTING

This testing is generally performed when the project is nearing its end. This test mainly qualifies the project and decides if it will be accepted by the users of the system. The users or the customers of the project are responsible for the test.

## 7.5 DATA VALIDATION TESTING

Data validation is the process of testing the accuracy of data; a set of rule you can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values into a form. In this project data validation testing carried out on all input form pages to test the accuracy.

## 7.6 WHITE BOX TESTING

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc. In this project there was many code errors occurred in many web forms, all errors were corrected through debugging.

## 7.7 BLACK BOX TESTING

Black Box Testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application.

## 7.8 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produces the required output generated or considered into two ways; one is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also the output comes out as the specified requirements by the user. Hence output testing does not result in any correction in the system.

## 7.9 TESTING RESULT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl No** | **Test Description** | **User** | **Expect Output** | **Actual Output** | **Result** |
| 1. | Send image | User | Send image if camera sense human motion | Image send | Pass |
| 2. | Capture photo | User | Capture photo if camera sense the human motion | Capture photos | Pass |
| 3. | Add new user | User | Registered successfully | Registered successfully | Pass |
| 4. | View user | User | Shows the list of users | Shows the list of users | Pass |
| 5. | View history | User | Shows the history of visitors | Shows visitors history | Pass |

Table 7.5: Testing and results

**CHAPTER 8**

# RESULTS AND DISCUSSION

The existing doorbell systems follow traditional approach, when a visitor presses the switch bell that rings inside the house. If someone is present in the house they open the door and if no one is present in the house the visitor waits for certain time and leaves the place without any clue. The proposed system using highly secured doorbell system overcomes the drawbacks of exiting system. Home security system using Raspberry pi. . When visitor motion detected at Door, Camera module interfaced to Raspberry Pi capture images, save it on system. When a visitor is at the door the user is notify via an app and user can authorized the visitor entry in the house. The concerned authority can control the system and view video stream of camera module through Smart mobile Phone. The system also provided concerned authority to use Smart Phone to send command for voice alert when intruder identified.

Table above in chapter 4 summarizes the results of tests conducted. The table gives the main characteristics of each action, its expected output and actual output obtained .The table gives

* The number of tests conducted  Description of the test.
* Who conducted the test
* What was the expected output
* What is the actual output
* Was is a pass or fail

## 8.1RESULTS

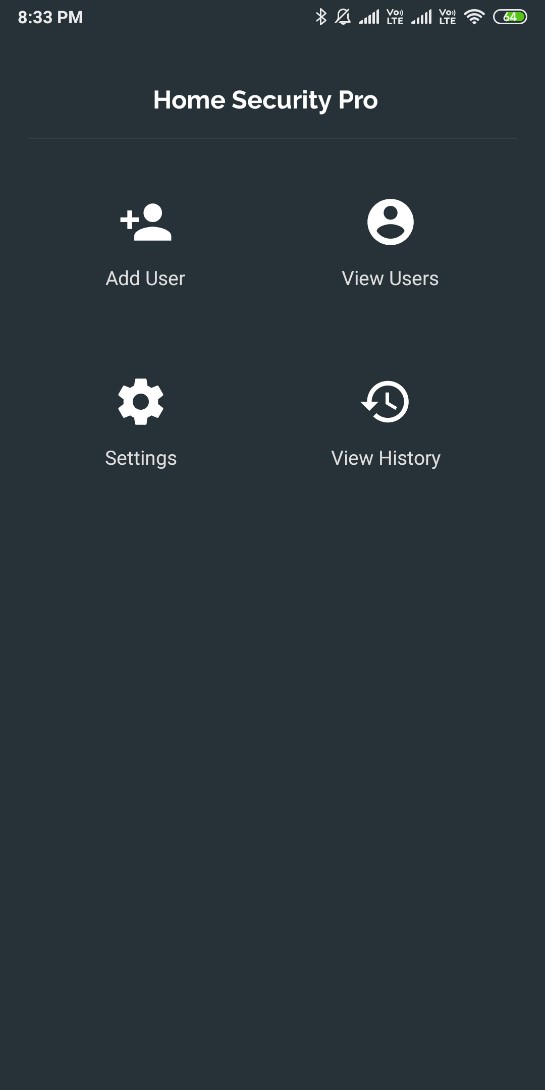
The proposed system incorporated with the following features.

* Quick and appropriate action can be taken easily for the user.
* Human effort can be reduced.
* Improved efficiency.
* Flexibility.
* Security enhanced for human life.
* Security

## 8.2 SCREEN SHOT

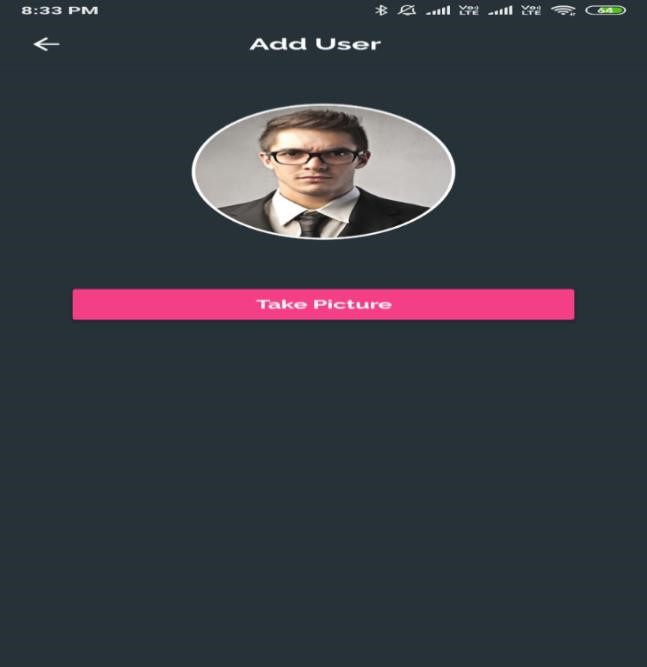
1. Android app main page

The following image represents the homepage/main page of the android application.



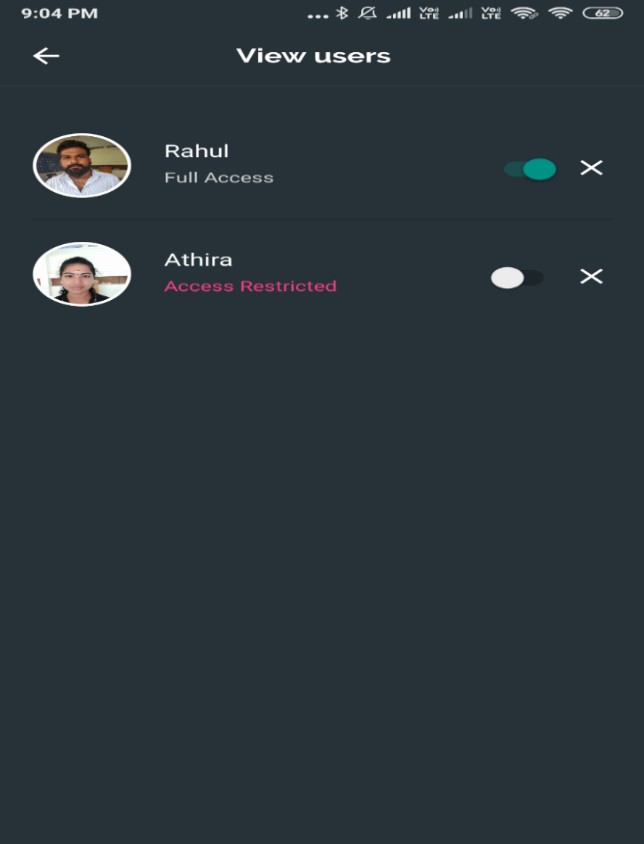
1. Registration page

The following image represents the registration page. Through this page owner can add the user with his image.



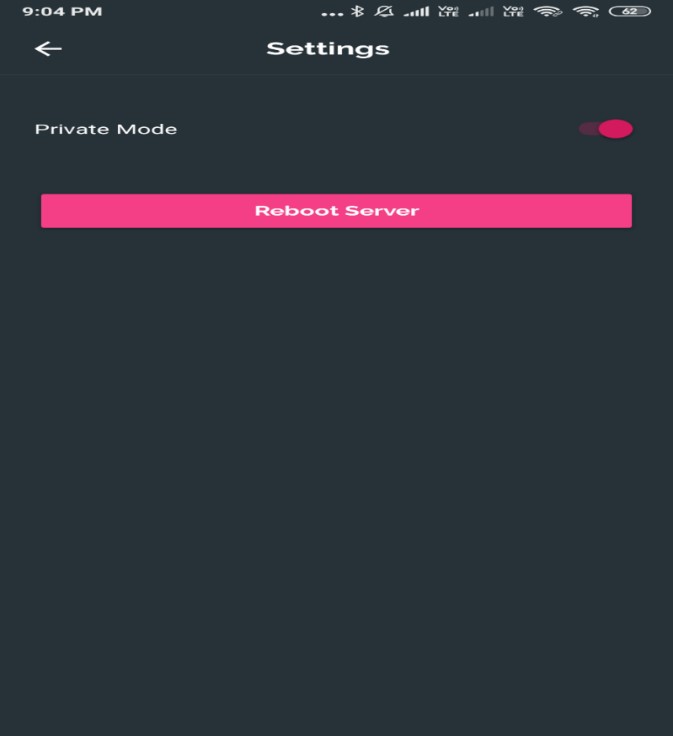
1. View users

The following image represents the view users page. Owner can view the list of registered users through this page.



1. Settings

The following image represents the settings page of the android application. This page helps the user to set the accessibility modes of user as private or public.



1. Face recognized

The following image represents the face recognition. The registered image is recognized by the server.



**CHAPTER 9**

# CONCLUSION

## 9.1 SYSTEM IMPLEMENTATION

Implementation means converting a new design into iteration .During implementation there should be a strong interaction between the developer of the software and the users.

Implementation involves installing hardware terminals and training the operating staff.

In this phase, user training is critical for minimizing reluctance to change and giving the new system a chance to prove its worth. The new system may be totally new replacing the existing system, or it may be the modifications of existing system. In either case proper implementation is essential to provide a reliable system to meet organizational requirements.

The project “HOME SECURITY SYSTEM” is an implementation of smart doorbell system and face recognition. The database will consists of captured images. Using deep learning algorithms the captured face is recognized. Thus, many components like pre-processing of image, classifiers, and prosodic features come together in the implementation of home security system.

## 9.2 CONCLUSION

The project has addressed the security issue of home. This project is a module which can be incorporated with any other system including hardware for building some fully automated machines. The software is designed in such a way that future modification can be done easily. The following conclusions can be deduced from the development of the project: It provides an abstract platform between the users and avoid manual work.

The experimental results of the system proposed indicate that it may provide a consistent support and assistance for safe and secure life with minimum cost of the system. Raspberry pi is a strong and reliable embedded system device for the complex and challenging tasks. Using these technologies in the proposed system will bring several advantages in providing safety and security for the owner and also comfortable for the visitors. With the use of Raspberry pi, camera and other various important modules, our homes are certainly more monitored and secured. This technology will definitely improve the security of our houses.

## 9.3 FUTURE ENHANCEMENT

* To improve with voice calling so that the visitor and the house owner could talk to each other and decide what to do
* To implement one-way or two-way audio and video communication between the visitor and the homeowner.

# BIBLIOGRAPHY

## 1. BOOKS

1. **Gary Bradski, Adrian Kaehler,** “Learning OpenCV: Computer Vision with the OpenCV

Library”,Stanford University.

1. **Towards Data Science** (2019). Doodling with Deep Learning!. [online] Available at:
2. **OpenCV** (2019). OpenCv—Python Core— [online]
3. **Woo-Hyun Park, Yun-Gyung Cheong** (2017) IOT Smart bell notification system: Design and Implementation,IEEE
4. **Sameer Deshpande, Anamika Majumdar, Atul Kumar Sahu , Vandita Gainedi, Deva Hema** , Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org Volume 8, Issue 4, April (2018)

## 2. WEBSITES

<http://engur.uconn.edu/~song/classes/nes/RPi.pdf> https://opencv.org/ [Accessed 9 May 2019].

<http://www.tutorialspoint.com/python_basic_syntax.ht> [http://www.w3schools.com](http://www.w3schools.com/)  [http://www.wikipedia.com](http://www.wikipedia.com/)  [https://stackoverflow.com](https://stackoverflow.com/)

https://towardsdatascience.com/doodling-with-deep-learning-1b0e11b858aa [Ac- cessed 9 May 2019].

I

# APPENDICES

## 1. SCRUM MODEL

### i. Git

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

### ii. Git Repositories

A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories.

If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user. Git also supports the creation of repositories targeting the usage on a server.

### iii. Scrum

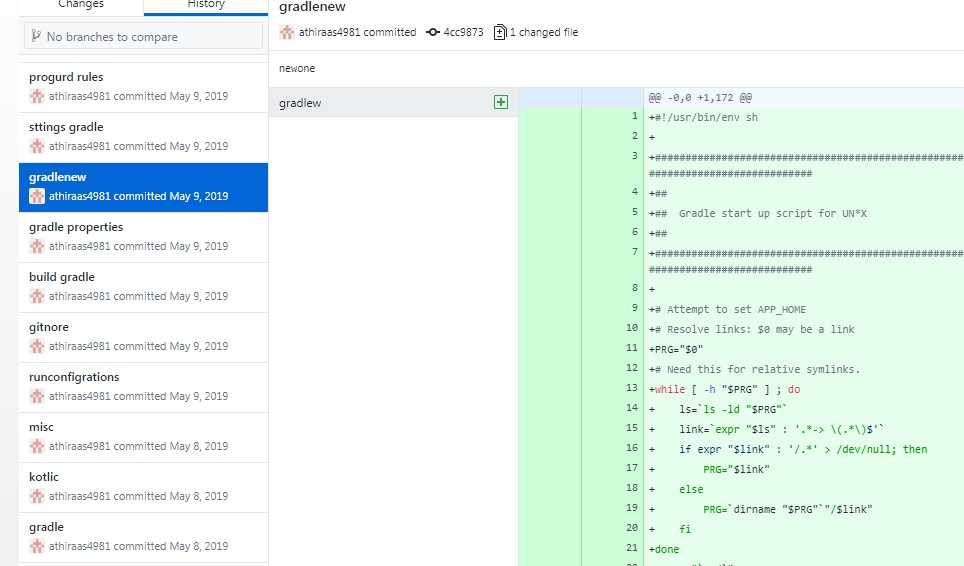
Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

II

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

### iv Git History



1. Git history

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page No** |
| 5.1 | Input output table | 18 |
| 7.1 | Unit test cases and results | 23 |
| 7.2 | Integration test cases and results | 24 |
| 7.3 | Testing and results | 26 |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name** | **Page No** |
| 5.1 | DFD Components | 14 |
| 5.2 | Context Diagram (level 0) | 15 |
| i | Android Application (Level-1) | 15 |
| ii | Android app main page | 28 |
| iii | Registration page | 29 |
| iv | View users | 29 |
| v | Settings | 30 |
| vi | Face recognized | 30 |
| 1 | Git history |  |

## ABBREVIATIONS AND NOTATION

### i. DFD

DFD (Data Flow Diagram) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

DFDs can also be used for the visualization of data processing (structured design).

### ii. GIT

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

### iii. DB

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

### vi. CODING

**1. deepface.py** import cv2 import requests import numpy as np import json import argparse import signal import logging import datetime, time from scipy import spatial import os from tinydb import TinyDB, Query import after\_response from pygame import mixer

from picamera.array import PiRGBArray from picamera import PiCamera

from gpiozero import LED from time import sleep

face\_api =

"http://192.168.43.192:5000/inferImage?returnFaceId=true&detector=yolo&returnFac eLandmarks=true"

# init logger

logger = logging.getLogger('Home pro security') logger.setLevel(logging.DEBUG)

# create console handler with a higher log level ch = logging.StreamHandler() ch.setLevel(logging.DEBUG)

# create formatter and add it to the handlers

formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s -

%(message)s') ch.setFormatter(formatter) # add the handlers to the logger logger.addHandler(ch)

# db db\_visitors = TinyDB('db/visitors.json')

#init sound library

#mixer.init()

# attendance register att\_reg = [] try:

att\_reg = json.loads(open('att\_log').read()) except: pass

led\_open = LED(18) led\_close = LED(4)

# parse arguments

parser = argparse.ArgumentParser(description='Home pro security System') parser.add\_argument('--w', action='store', default=320, nargs='?', help='Set video width')

parser.add\_argument('--h', action='store', default=240, nargs='?', help='Set video height') args = parser.parse\_args()

# initialize database db = {"names":[],"embeddings":[]} dbtree = "" try:

db = json.loads(open('face\_data.txt').read()) dbtree = spatial.KDTree(db["embeddings"]) except: pass

# start the camera

# cap = cv2.VideoCapture(args.src)

# cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, float(args.w))

# cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, float(args.h))

# cap.set(cv2.CAP\_PROP\_FPS, 10)

camera = PiCamera() camera.resolution = (320, 240) camera.framerate = 20 rawCapture = PiRGBArray(camera, size=(320, 240)) time.sleep(0.2)

# search for a face in the db def identify\_face(embedding): if dbtree != "":

dist, idx = dbtree.query(embedding) name = db["names"][idx] print(name) if dist > 0.5:

name = "unknown" else:

name = "unknown"

return name

def play\_sound(name): print("playing audio...") os.system("mpg123 "+ "sounds/" + name + ".mp3") #mixer.music.load("sounds/" + name + ".mp3")

#mixer.music.play()

# returns minutes since def mins\_since\_last\_log():

return ((datetime.datetime.now() - datetime.datetime.strptime(att\_reg[-1]['time'],

'%Y-%m-%d %H:%M:%S')).seconds/60)

def mark\_present(name): if len(att\_reg) == 0:

logger.info("Detected %s"%name) stime = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d

%H:%M:%S')

att = {'name':name,'time':stime} att\_reg.append(att) return

if att\_reg[-1]['name'] != name or mins\_since\_last\_log() > 1:

logger.info("Detected %s"%name) stime = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d

%H:%M:%S')

att = {'name':name,'time':stime} att\_reg.append(att)

# door close and open functions def door\_open(): led\_open.on() led\_close.off()

def door\_close(): led\_close.on() led\_open.off() # start processing count = 0 user\_count = 0; prev\_name = [] door\_close() while True:

for frame in camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True):

rawCapture.truncate(0) image = frame.array #cv2.imshow("Frame", image) key = cv2.waitKey(1) & 0xFF

framex = image key = cv2.waitKey(1) & 0xFF

count += 1 if count % 2 != 0:

continue

frame = cv2.resize(framex, (int(args.w),int(args.h)))

r, imgbuf = cv2.imencode(".bmp", frame) image = {'pic':bytearray(imgbuf)}

r = requests.post(face\_api, files=image) result = r.json()

if len(result) > 1: faces = result[:-1] diag = result[-1]['diagnostics'] print(user\_count) for face in faces:

rect, embedding = [face[i] for i in ['faceRectangle','faceEmbeddings']] x,y,w,h, confidence = [rect[i] for i in ['left', 'top', 'width', 'height', 'confidence']]

if confidence < 0.8: user\_count = 0 prev\_name = [] continue

name = identify\_face(embedding) if(name not in prev\_name): prev\_name.append(name)

if name == "unknown":

user\_count += 1

else: if name != "unknown":

user\_count += 1

mark\_present(name) if(name not in prev\_name):

user\_count = 0 prev\_name = []

if user\_count > 2: print("Found some one => ", prev\_name) ts = time.time() url = str(ts) + '.jpg'

dt = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d

%H:%M:%S')

st = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

cv2.imwrite('visitors/' + url, frame) db\_visitors.insert({'name': prev\_name, 'url': url, 'date': dt, 'time': st}) print("Inserted")

visitors = ", ".join(prev\_name)

# read settings

db\_settings = TinyDB('db/settings.json') db\_settings = TinyDB('db/settings.json') settings\_list = db\_settings.all(); private\_mode = settings\_list[0]['private'];

# read users list name\_list = []

db\_users = TinyDB('db/users.json') users\_list = db\_users.all(); for item in users\_list:

name\_list.append(item)

if(private\_mode == False):

if( "unknown" not in prev\_name):

for itx in prev\_name: access\_flag = True; users\_list = db\_users.all(); item\_index = 0; for item in users\_list:

if(item['name'] == itx and users\_list[item\_index]['access'] == False):

access\_flag = False

break item\_index += 1 if(access\_flag == False):

break

if(access\_flag):

play\_sound("granted") print("door is opening") door\_open() else:

play\_sound("denied") print("You have no permission to open")

else: print("Please wait for approval") else:

play\_sound("denied") print("aceess denied!")

after\_response.send\_push(visitors, st)

prev\_name = [] user\_count = 0 time.sleep(5) door\_close()

cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,255),5,8) cv2.rectangle(frame, (x,y+h-20), (x+w,y+h), (255,0,255), -1, 8)

|  |  |
| --- | --- |
| cv2.putText(frame, "%s"%(name), cv2.FONT\_HERSHEY\_DUPLEX, 1, (255,255,255),2,8) | (x,y+h), |
| cv2.putText(frame, diag['elapsedTime'], | (0,20), |

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0,0,255))

cv2.imshow("Home Pro Security System", frame) if key == ord('q'):

break

print("Exit")

#### i. App

<?xml version="1.0" encoding="UTF-8"?>

<manifest package="**com.example.arunjith.homeprosecurity**"

xmlns:android="**http://schemas.android.com/apk/res/android**"><uses-permission android:name="**android.permission.INTERNET**"/><uses-feature android:name="**android.hardware.camera**" android:required="**true**"/><uses-permission android:name="**android.permission.WRITE\_EXTERNAL\_STORAGE**"/><application android:theme="**@style/AppTheme**" android:supportsRtl="**true**"

android:roundIcon="**@mipmap/ic\_launcher\_round**" android:label="**@string/app\_name**" android:icon="**@mipmap/ic\_launcher**" android:allowBackup="**true**"><activity android:name="**.MainActivity**"/><provider android:name="**android.support.v4.content.FileProvider**"

android:grantUriPermissions="**true**" android:exported="**false**"

android:authorities="**com.example.android.fileprovider**"><meta-data android:name="**android.support.FILE\_PROVIDER\_PATHS**" android:resource="**@xml/file\_paths**"/></provider><activity android:name="**.HomeActivity**" android:theme="**@style/AppTheme.NoActionBar**" android:label="**@string/app\_name**"><intent-filter><action android:name="**android.intent.action.MAIN**"/><category android:name="**android.intent.category.LAUNCHER**"/></intent-

filter></activity><meta-data android:name="**preloaded\_fonts**"

android:resource="**@array/preloaded\_fonts**"/><activity android:name="**.ViewUsers**"/><activity android:name="**.Settings**"/><activity android:name="**.Visitors**" android:theme="**@style/AppTheme**"

android:label="**@string/title\_activity\_visitors**"/></application></manifest>

<?xml version="1.0" encoding="UTF-8"?>

<project version="**4**"><component name="**WizardSettings**"><option name="**children**"><map><entry key="**vectorWizard**"><value><PersistentState><option name="**children**"><map><entry key="**vectorAssetStep**"><value><PersistentState><option name="**children**"><map><entry key="**clipartAsset**"><value><PersistentState><option name="**values**"><map><entry key="**url**" value="**jar:file:/Applications/Android%20Studio.app/Contents/plugins/android/li b/android.jar!/images/material\_design\_icons/image/ic\_edit\_black\_24dp.xml**"/></ map></option></PersistentState></value></entry></map></option><option

#### name="values"><map><entry key="color" value="ffffff"/><entry key="outputName" value="ic\_edit\_black\_24dp"/><entry key="overrideSize" value="true"/><entry key="sourceFile"

value="**$USER\_HOME$**"/></map></option></PersistentState></value></entry></m ap></option></PersistentState></value></entry></map></option></component></pr oject>