

Face Recognition and Attendance System

Submitted by

Prajwal K Augustine

182010

Msc Big Data Analytics
AIMIT, St. Aloysius College
Mangalore

Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Big Data Analytics (MSc)

Under the guidance of

Mrs. Laveena C Crasta

HOD, Department of Big Data Analytics
AIMIT, St. Aloysius College
Mangalore

Submitted to



**ST ALOYSIUS INSTITUTE OF MANAGEMENT AND
INFORMATION TECHNOLOGY (AIMIT)
ST ALOYSIUS COLLEGE (AUTONOMOUS)
MANGALURU, KARNATAKA**

2020



**ST ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY**
ST ALOYSIUS COLLEGE (AUTONOMOUS)
MANGALORE, KARNATAKA

CERTIFICATE

This is to certify that the project titled

Face Recognition and Attendance System

Submitted by

Prajwal K Augustine
182010
Msc Big Data Analytics
AIMIT, St. Aloysis College
Mangalore

*In partial fulfillment of the requirements for award of degree of Master of big data analytics of Mangalore University, is a bonafide record of the work carried out at **Ihub IT Solutions, Mangalore***

During the year
2020

Mrs. Laveena C Crasta
HOD
Department of Big Data Analytics
AIMIT, St. Aloysis College,
Mangaluru-575 022.

Prof Santhosh Rebello
Dean, School of IT
PG Dept of Information Technology,
AIMIT, St. Aloysis College,
Mangaluru-575 022.

Rev. Dr. Fr. Melvin Pinto SJ
Director,
St. Aloysis College
Mangaluru-575 022.

Examiners

1.

2.



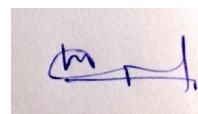
**ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE - 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY**

PROFORMA OF PROJECT PROPOSAL

Note : The student has to fill this form and approval has to be sought from the Dean of the faculty before starting the Industry Internship.(January I week)

- [1]. Student Register Number : 182010
- [2]. Student Name : Prajwal K Augustine
- [3]. Title of the Project : Face Recognition and Attendance System
- [4]. Name & Address of the Organization :
iHub IT Solutions, First Floor, Soorya Commercials, Near Soorya Infratech Park, Mudipu, Kurnadu,Karnataka, 574153
- [5]. Telephone No. : 9562781634
- [6]. E-mail : info@ihubonline.co / ihubitsolutions@gmail.com
- [7]. Website / URL : www.ihubonline.co
- [8]. Address of the Project Site : iHub IT Solutions
- [9]. Telephone No. : 9562781634
- [10]. E-mail : info@ihubonline.co
- [11]. Residential Address of the Student : "Divine Grace" Jangamarabettu Kallianpur Udupi Karnataka 576114
- [12]. Telephone No : NIL
- [13]. Mobile Number : 9741193484
- [14]. E-mail : prajwalaugustine97@gmail.com

I hereby declare that the above furnished details are true to best of my knowledge and request you to permit me to join the Industry Internship for the Project work prescribed for MSc Big Data Analytics.



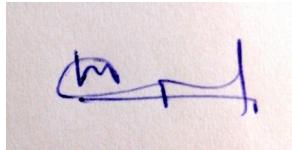
Dated : Signature of the Student

The Proposal for the project work has been accepted

Dean of the Faculty

CERTIFICATE OF AUTHENTICATED WORK

This is to certify that the domain seminar project report entitled **Face Recognition and Attendance System** submitted to Aloysis Institute of Management and Information Technology (AIMIT), St Aloysis College, Mangalore affiliated to Mangalore University in partial fulfilment of the requirement for the award of the degree of MASTER OF SCIENCE (MSC BIG DATA ANALYTICS) is an original work carried out by **Mr. Prajwal K Augustine**, Register no. **182010** under my guidance. The matter embodied in this domain seminar project is authentic and is genuine work done by the student and has not been submitted whether to this University, or to any other University / Institute for the fulfilment of the requirement of any course of study.



Signature of the Student:

Date: 15 / 06 / 2020

Prajwal K Augustine
182010

“Divine Grace”
Jangamarabettu Kallianpur
Udupi, 576114



Signature of the Guide

Date: 15 / 06 / 2020

Mrs. Rekha R

Project Manager
Soorya Infratech Park,
Mudipu,
Mangalore Karnataka
574153
www.ihubonline.co
info@ihubonline.co

www.ihubonline.co

Ref: Ihub0172020J
13 - Jun -2020

TO WHOMSOEVER IT MAY BE COCERNED

This is to certify that **Mr. Prajwal K Augustine (Reg No : 182010)**, student from **MSc Big Data Analytics, St. Aloysius Institute of Management and Information Technology (AIMIT) Beeri , Mangalore** has successfully completed the Internship Program on Data Science (Machine learning and Deep learning) from 6th January 2020 for a duration of 5 months.

During this period he was observed to be hard working and a good team player. His association with us was very fruitful and we wish him all the best in his future endeavors.

Sincerely,

Rekha R
Project Manager
Soorya Infratech Park, Mudipu,
Mangalore Karnataka 574153
www.ihubonline.co info@ihubonline.co



iHub ITSolutions

Mail: info@ihubonline.co

www.ihubonline.co

Innovate Automate Transform

ABSTRACT

Face Recognition and Attendance System is a Deep Learning Based Computer Vision system which can be used to identify faces and retrieve information. The system uses the Deep Convolutional Neural Network Model Framework in tensorflow environment. The Model uses the FaceNet model architecture which is a custom deep neural network with specific layers to increase the accuracy of the classification process. The security staff can use this system to scan the faces of their suspects and retrieve information on them. If the suspect is authorized the system will retrieve information of that employee from the database. If the suspect is not authorized he will be apprehended and the security staff will be alerted.

Project Proposal Report / Synopsis

I. Title of the Project

The title of the project is “**Face Recognition and Attendance System**”

II. Statement of the Problem

The Security Staff of a large Enterprise are staged at many places across the company. Sometimes unauthorized persons enter the building and go unnoticed. They might pose a threat to the company. Even if the said person is spotted the Security Staff cannot apprehend the person without authorization without a confirmation from a superior officer. This delay might result in a bad incident.

III. Why this particular topic chosen?

This topic mainly explores the domain of computer vision which has a lot of scope in the current market. Facial recognition is a part of computer vision which is still not up to the mark. The various methods used in facial recognition is inaccurate as time progresses. But with the advancements in deep learning algorithms such as CNN, RCNN etc we can build advanced facial regression system which can detect and recognize a face in a much better way.

IV. Objective and Scope

The end goal of this project is to build a system which can monitor the employees of a firm and their various activities. The employees will be recognized and the attendance will be

automatically recorded, which will give also some weekly visualizations. The system will identify unauthorized users who enter the firm. It can classify the users as trusted or a threat. If the user is a threat the security of the firm will be deployed to that particular location.

V. Methodology

The methodology we used here is Agile, because agile software development allows the team to work together more efficiently and effectively in developing complex projects. It consists of practices that exercise iterative and incremental techniques which are easily adopted and display great results.

The objective of using this methodology is to address the following:

- Individuals and interactions: in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- Working software: working software is more useful and better than just presenting documents to clients in meetings.
- Customer collaboration: requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.
- Responding to change: agile development is focused on quick responses to change and continuous development.

VI. Process Description

Facial Recognition Module :

- This module can be accessed by both the Security Staff and Admin.
- It is used to identify a employee and display his details
- The Neural Network for this module is trained using python in a tensorflow environment.

- The Neural Network follows the FaceNet configuration
- The Neural Network needs A GPU or TPU to run.
- The image of the employee will be classified using this Neural Network and the details will be retrieved from the data base.

Attendance Module :

- This module can be accessed by Admin, Security Staff and the Employee
- Is used to punch in the attendance based on facial identification
- The Neural Network for this module is trained using python in a tensorflow environment.
- The Neural Network follows the FaceNet configuration
- The Neural Network needs A GPU or TPU to run.
- The image of the employee will be classified using this Neural Network and the attendance will be punched in.

VII. Resources and Limitations

Hardware requirements:

Processor: Intel® Core™ i3-6006U CPU

RAM: 8 GB or above

Hard disk: 500 GB or above

Software requirements:

Operating System: Microsoft Windows 2010 (32 bit and 64 bit) or above.

Browser version: Microsoft Internet Explorer 8.0 or above versions, Google Chrome.

Technology used: Python, Flaks, Google Collab.

VIII. Testing Technologies used

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any

gaps, errors, or missing requirements in contrary to the actual requirements. The testing techniques used in this project are:

- **Unit Testing :**

A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. Testing is done by checking for all the boundary conditions, path specification, error conditions, null values and whether the information is properly flowing in to the program unit (or module) and properly happen out of it or not etc.

- **Integration Testing :**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

- **System Testing :**

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification(s).

- **Compatibility Testing :**

Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.

Compatibility Testing is a type of Non-functional testing

Types of Compatibility Testing:

- **Hardware :** It checks software to be compatible with different hardware configurations.

- Operating Systems : It checks your software to be compatible with different operating systems like Windows, UNIX, and Mac OS etc.
- Software : It checks your developed software to be compatible with other software.
- Network : Evaluation of performance of a system in a network with varying parameters such as Bandwidth, Operating speed, Capacity. It also checks application in different networks with all parameters mentioned earlier.
- Browser : It checks the compatibility of your website with different browsers like Firefox, Google Chrome, and Internet Explorer etc.
- Devices : It checks compatibility of your software with different devices like
- USB port Devices, Printers and Scanners, Other media devices and Bluetooth.
- Mobile : Checking your software is compatible with mobile platforms like Android, iOS etc.
- Versions of the software : It is verifying your software application to be compatible with different versions of the software. For instance checking your Microsoft Word to be compatible with Windows 7, Windows 7 SP1, Windows 7 SP2, and Windows 7 SP3.
- **Regression Testing :**

Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the new code changes are done.

IX. Conclusion

This system focuses on improving the productivity and security of the organization by monitoring its employees and making sure the work environment is safe.

ACKNOWLEDGMENT

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success. Especially, please allow me to dedicate my acknowledgment of gratitude towards the following significant advisers and contributors:

I am very grateful to **Ihub IT Solutions, Mangalore** for having provided me this project with necessary facilities and a good environment. I would like to thank my manager **Mrs. Rekha R** for his guidance and the motivation provided during this project period. I would like to convey my gratitude to **Mr. Vipin** for building up presentation skills and confidence into a whole new level.

I would like to thank my internal guide **Mrs. Laveena C Crasta** who has provided me with all valuable inputs to put my efforts on the right track.

I would like to convey my gratitude to **Fr. Denzil Lobo SJ** for being a pillar of support in all my endeavors.

I would like to thank **Rev. Dr. Praveen Martis SJ**, Principal of St Aloysius College and **Rev. Fr. Melwyn Pinto SJ**, Director of Aloysius Institute of Management and Technology (AIMIT) for their constant support and encouragement.

I would also like to thank **Prof Santhosh Rebello** the Dean, PG Department of Information technology for his great support, and instruction rendered to me during my project work.

Finally, I sincerely thank my parents, family, and friends, who have been a constant source of inspiration during the preparation of this project work

Place : Mangaluru

Date : 15 / 06 / 2020

TABLE OF CONTENTS

| | |
|---|-----------|
| Chapter 1 : INTRODUCTION..... | 18 |
| 1.1 Background..... | 19 |
| 1.2 Objectives..... | 19 |
| 1.3 Purpose,Scope and Applicability..... | 19 |
| 1.3.1 Purpose..... | 19 |
| 1.3.2 Scope..... | 19 |
| 1.3.3 Applicability..... | 19 |
| 1.4 Achievements..... | 20 |
| 1.5 Organisation Report..... | 20 |
| | |
| Chapter 2 : SURVEY OF TECHNOLOGIES..... | 22 |
| 2.1 Deep Learning | 23 |
| 2.2 Computer Vision..... | 23 |
| 2.3 Face Detection..... | 23 |
| 2.4 Face Recognition..... | 24 |
| 2.4.1 Geometric Based / Template Based..... | 24 |
| 2.4.2 Piecemeal / wholistic..... | 25 |
| 2.4.3 Appearance-Based / Model Based..... | 25 |
| 2.4.4 Template / Statistical / Neural Networks Based..... | 25 |
| 2.5 Flask..... | 26 |
| 2.6 Flutter..... | 26 |
| 2.7 Mongo DB..... | 26 |
| 2.8 Perceptron..... | 27 |
| 2.9 Convolutional Neural Network..... | 28 |
| 3.0 Tensorflow..... | 29 |
| 3.1 Keras..... | 29 |

| | |
|---|-----------|
| 3.2 FaceNet..... | 29 |
| Chapter 3 : REQUIREMENTS AND ANALYSIS..... | 32 |
| 3.1 Problem Definition..... | 33 |
| 3.2 Requirement Specification..... | 33 |
| 3.2.1 Stable Internet Connection..... | 33 |
| 3.2.2 Software Quality Attributes..... | 33 |
| 3.2.2.1 Reliability..... | 33 |
| 3.2.2.2 Availability..... | 33 |
| 3.3 Planning and Scheduling..... | 33 |
| 3.4 Software and Hardware Requirements..... | 36 |
| 3.5 Preliminary Product Description..... | 37 |
| 3.6 Conceptual Model..... | 37 |
| 3.6.1 Data Flow Diagram..... | 37 |
| Chapter 4 : SYSTEM DESIGN..... | 40 |
| 4.1 Basic Models..... | 41 |
| 4.1.1 Face Recognition Module..... | 41 |
| 4.1.2 Attendance Module..... | 41 |
| 4.2 Data Design Project Structure..... | 42 |
| 4.2.1 Schema Design..... | 42 |
| 4.3 Procedural Design..... | 43 |
| 4.3.1 Logic Diagrams | 43 |
| 4.3.1.1 Sequence Diagram..... | 43 |
| 4.3.1.2 Use Case Diagram..... | 46 |
| 4.3.1.3 Activity Diagram..... | 48 |
| 4.3.1.4 State Chart Diagram..... | 51 |
| 4.3.1.5 Class Diagram | 42 |
| 4.4 User Interface Diagram..... | 54 |

| | |
|--|-----------|
| 4.5 Security Issues..... | 58 |
| 4.6 Test Case Design..... | 58 |
| Chapter 5 : IMPLEMENTATION AND TESTING..... | 60 |
| 5.1 Implementation Approaches..... | 61 |
| 5.2 Coding Details and Code Efficiency..... | 61 |
| 5.3 Testing Approach..... | 62 |
| 5.3.1 Unit Testing..... | 62 |
| 5.3.2 Integration Testing..... | 62 |
| 5.3.3 User Acceptance Testing..... | 62 |
| 5.3.4 Performance Testing..... | 63 |
| 5.3.5 Load Testing/ Stress Testing..... | 63 |
| 5.4 Modification And Improvement..... | 63 |
| Chapter 6 : RESULTS AND DISCUSSION..... | 64 |
| 6.1 Test Reports..... | 65 |
| 6.2 User Documentation..... | 66 |
| Chapter 7 : CONCLUSION..... | 67 |
| 7.1 Conclusion..... | 68 |
| 7.2 Limitations..... | 68 |
| 7.3 Future Scope of the project..... | 68 |
| Chapter 8: REFERENCES..... | 69 |
| Chapter 9: GLOSSARY..... | 71 |

Table of Figures

| | |
|--|----|
| Figure 1 .Zero level DFD for Face Recognition and Attendance System..... | 38 |
| Figure 2.Level 1 DFD for Face Recognition and Attendance System..... | 39 |
| Figure 3. Sequence Diagram for Face Recognition..... | 44 |
| Figure 4. Sequence Diagram For Attendance System..... | 45 |
| Figure 5. Use case Diagram for Face Recognition and Attendance System..... | 47 |
| Figure 6. Activity Diagram for Face Recognition..... | 49 |
| Figure 7. Activity Diagram for Attendance System..... | 50 |
| Figure 8. State Chart Diagram for Face Recognition..... | 51 |
| Figure 9. State Chart Diagram for Attendance System..... | 52 |

List of Tables

| | |
|--|----|
| Tabel 1 : Database Table for employee..... | 42 |
| Tabel 2 : Database Table for attendance..... | 42 |
| Tabel 3 : Database Table for visistor | 43 |

CHAPTER 1 : INTRODUCTION

1.1 Background

The goal is not to provide a general of “what makes a good design”. Design is a creative activity that cannot be done mechanically; it requires human insight. The design activity is a fundamental phase in a process that progressively transforms the system requirements through a number of intermediate stages into final product. Software design is defined as a decomposition of the system into module description of what each module is intended to do and of the relationship among the modules. The decomposition of system into modules can be accomplished in several ways and in several steps. When system is decomposed into higher-level modules then such systems are called sub systems. Relations among the sub systems are then defined, and designers agree upon the intended behaviour of each sub system. Next each system is analysed separately, and the procedure is iterated until the complexity of each component is sufficiently that a single person can implement it readily.

1.2 Objectives

The staff should be able to scan the face of the suspect and get immediate results so that he can choose to clear or apprehend the suspect. If the suspect is unauthorized the neural network should classify it as unauthorized and alert all the staff in the company about that particular suspect. The attendance system should be able to recognize the person with high accuracy even when he grows some facial hair or when he is wearing some kind of objects like glasses which might not give a full read of the face.

1.3 Purpose , Scope And Applicability

1.3.1 Purpose

The purpose of this document is to fully document the software requirements using the system design document format in accordance with IEEE standard. It can also be used as a guide to user on how to use the software effectively.

1.3.2 Scope

The design activity is often divided into separate phase system design is a sometimes also called top level design. At the end of the system design all the major data structure file format and the major modules in the system and their specification are decided.

1.3.3 Applicability

This system is operated by the Security Staff and Admin.

1.4 Achievements

Undertaking the above project, personally has given me opportunity to learn completely new set of technologies and concepts which I was unaware before. Also, has helped me understand process better within the organization. This Project makes use of Flutter and MongoDB which helps in making dynamic application. We learn how to write a code that can handle situation efficiently. I learnt to write optimized code. I also learnt the importance of each line of code that we write that affect the system. This project has introduced me to a lot of deep learning algorithms and the advanced concepts of computer vision. I'm now familiar with working on neural networks in the tensorflow environment.

1.5 Organisation of Report

➤ Chapter 1: Introduction

The system being developed is introduced in this chapter. A description of the background and context of the project is specified. Concise statement of the aims and objectives of the project along with the purpose, scope and applicability are mentioned.

➤ Chapter 2: Survey of technologies

This chapter provides a detail of all the technologies which are used to complete the project. This also demonstrates the awareness and understanding of the available technologies related to the topic of the project.

➤ Chapter 3: Requirements and Analysis

The requirements of that lead to the development of the system are mentioned and analyzed in this chapter. This includes the problem definition, requirement specification and planning and scheduling. The details of the overall problem and the sub problems are provided in the problem definition phase. Planning for the purposes and the certain rules and constraints are specified in the planning and scheduling phase. The hardware requirements and software requirements are specified in the document. The hardware required to run the product in the system is clearly specified. The software requirements such as operating system and other software to link and install the software are listed in this section. The requirements and objectives of the new system, its functions and operations are given in the product description

section. The conceptual models are built to understand the problem domain which describes the operations that can be performed and the allowable sequences of those operations.

➤ **Chapter 4: System Design**

System Design describes the features and operations in detail, including the data design, functional and interface design, process diagrams, details description of functions and the test case design. The Basic Modules section briefly describes all the modules and the functionality of these modules. The next section is the Data Design which consists of how to organize, manage and manipulate the data. This section again has two Subsections-Schema design and Data Integrity and constraints. Schema design defines the structure and explanation of schemas. Data Integrity and constraints specifies all the validity checks and constraints.

➤ **Chapter 5: Implementation and testing**

This chapter defines the plan of implementation and standards used in implementation, code efficiency, and testing approach, integrated testing, Modifications and improvements of project.

➤ **Chapter 6: Results and Decisions**

Here the test reports explain the test results and reports based on the test cases thereby showing that the software is capable of facing any problematic situation and that it works fine in different conditions. And the user documentation defines the working of the software i.e. explaining its different functions, components with screen shots.

➤ **Chapter 7: Conclusions**

This chapter summarizes the entire work. It explains all the points that have been made in the earlier chapters. It also includes the limitations of the system, listing all the criticism encountered during the demonstration of the software. It also proposes the future scope of the project i.e. new areas of investigation prompted by developments. It also explains the limitations of the system and the future possible enhancements of the system.

CHAPTER 2 : SURVEY OF TECHNOLOGIES

2.1 Deep Learning

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

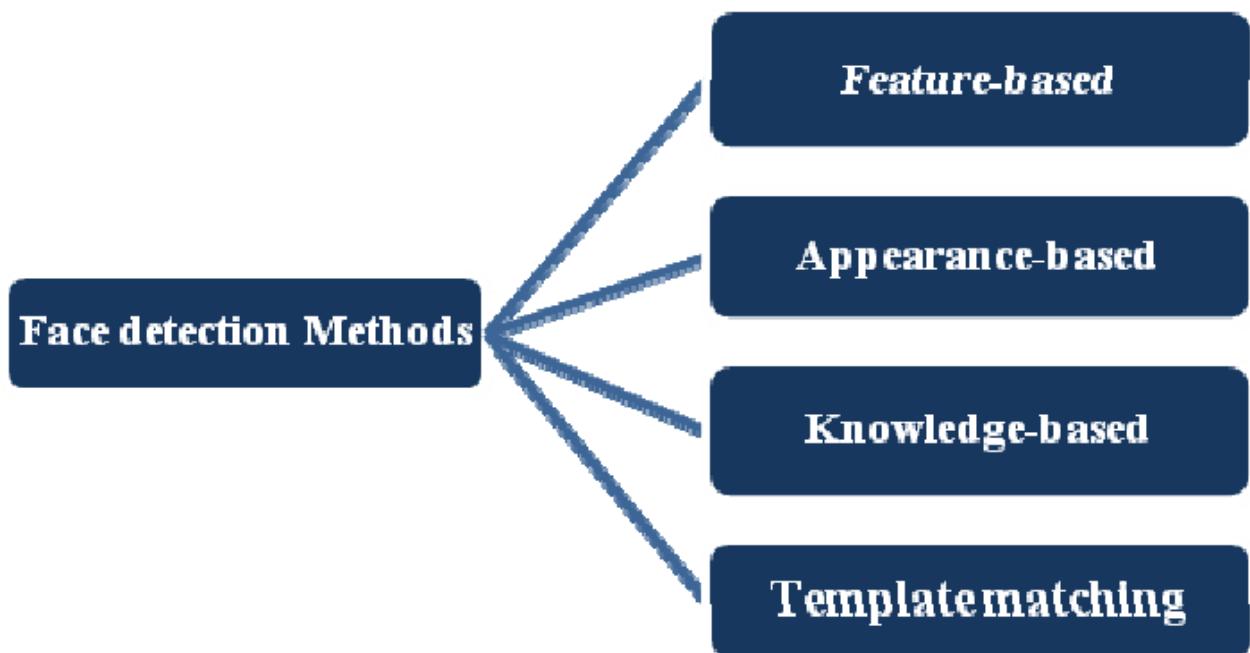
2.2 Computer vision

Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do. Until recently, computer vision only worked in limited capacity. Thanks to advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to take great leaps in recent years and has been able to surpass humans in some tasks related to detecting and labeling objects. One of the driving factors behind the growth of computer vision is the amount of data we generate today that is then used to train and make computer vision better.

2.3 Face Detection

In the past few years, face recognition owned significant consideration and appreciated as one of the most promising applications in the field of image analysis. Face detection can consider a substantial part of face recognition operations. According to its strength to focus computational

resources on the section of an image holding a face. The method of face detection in pictures is complicated because of variability present across human faces such as pose, expression, position and orientation, skin colour, the presence of glasses or facial hair, differences in camera gain, lighting conditions, and image resolution. Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. It is a part of object detection and can be used in many areas such as security, bio-metrics, law enforcement, entertainment, personal safety, etc.



2.4 Face Recognition

Face Recognition is a recognition technique used to detect faces of individuals whose images are saved in the data set. Despite the point that other methods of identification can be more accurate, face recognition has always remained a significant focus of research because of its non-meddling nature and because it is people's facile method of personal identification.

2.4.1 Geometric Based / Template Based

Face recognition algorithms classified as geometry based or template based algorithms. The template-based methods can be constructed using statistical tools like SVM [Support Vector

Machines], PCA [Principal Component Analysis], LDA [Linear Discriminant Analysis], Kernel methods or Trace Transforms. The geometric feature based methods analyse local facial features and their geometric relationship. It is also known as a feature-based method.

2.4.2 Piecemeal / Wholistic

The relation between the elements or the connection of a function with the whole face not undergone into the amount, many researchers followed this approach, trying to deduce the most relevant characteristics. Some methods attempted to use the eyes, a combination of features and so on. Some Hidden Markov Model methods also fall into this category, and feature processing is very famous in face recognition.

2.4.3 Appearance-Based / Model-Based

The appearance-based method shows a face regarding several images. An image considered as a high dimensional vector. This technique is usually used to derive a feature space from the image division. The sample image compared to the training set. On the other hand, the model-based approach tries to model a face. The new sample implemented to the model and the parameters of the model used to recognise the image.

The appearance-based method can classify as linear or nonlinear. Ex- PCA, LDA, IDA used in direct approach whereas Kernel PCA used in nonlinear approach. On the other hand, in the model-based method can be classified as 2D or 3D Ex- Elastic Bunch Graph Matching used.

2.4.4 Template / Statistical / Neural Networks Based

In template matching the patterns are represented by samples, models, pixels, textures, etc. The recognition function is usually a correlation or distance measure. In the Statistical approach, the patterns expressed as features. The recognition function in a discriminant function. Each image represented regarding d features. Therefore, the goal is to choose and apply the right statistical tool for extraction and analysis. Neural Network has continued to use pattern recognition and classification. Kohonen was the first to show that a neuron network could be used to recognise aligned and normalised faces. There are methods, which perform feature extraction using neural networks. There are many methods, which combined with tools like PCA or LCA

and make a hybrid classifier for face recognition. These are like Feed Forward Neural Network with additional bias, Self-Organizing Maps with PCA, and Convolutional Neural Networks with multi-layer perception, etc. These can increase the efficiency of the models.

2.5 Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

2.6 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Windows, Mac, Linux, Google Fuchsia and the web. Flutter apps are written in the Dart language and make use of many of the language's more advanced features. On Windows, macOS and Linux via the semi-official Flutter Desktop Embedding project, Flutter runs in the Dart virtual machine which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code can be reflected immediately in the running app without requiring a restart or any loss of state.

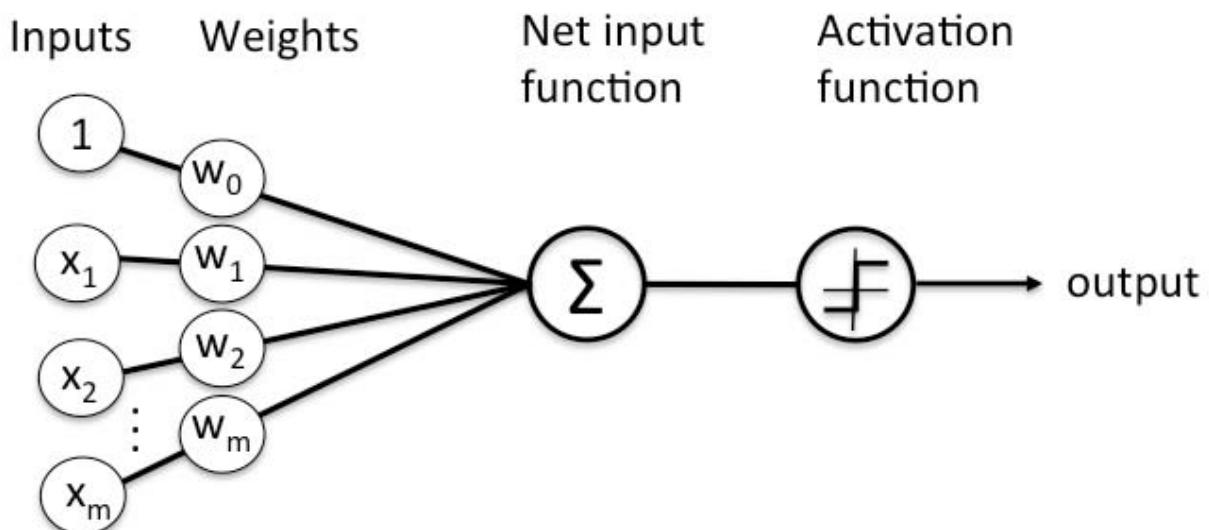
2.7 MongoDB

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database. MongoDB is a

document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model.

2.8 Perceptron

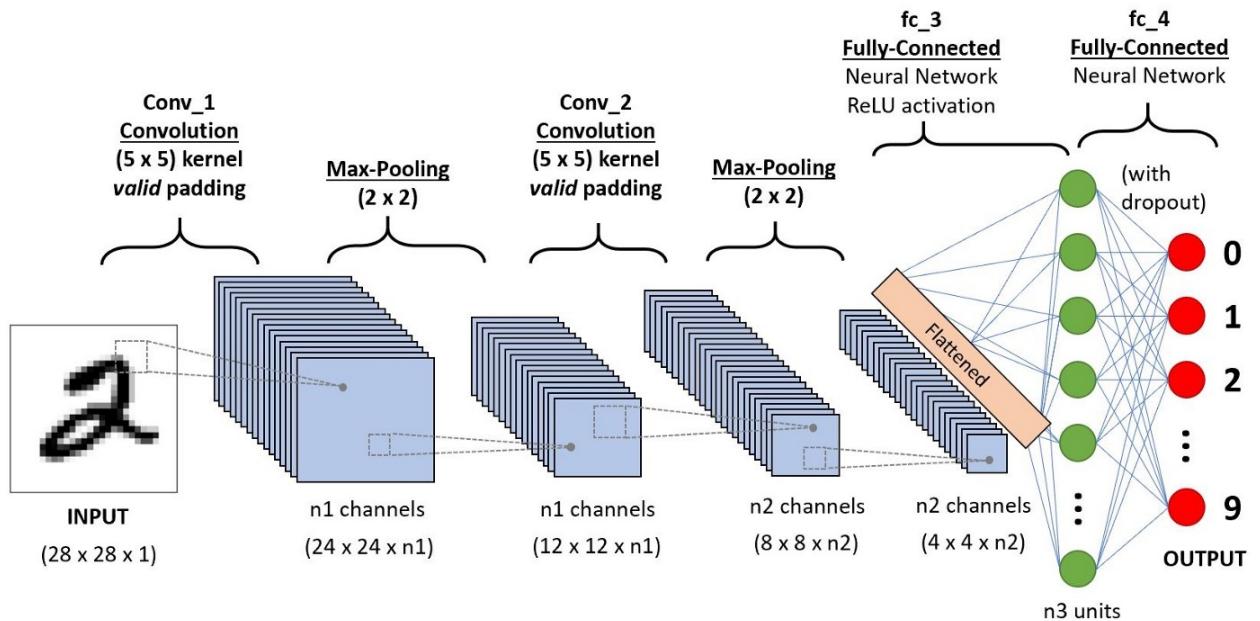
A perceptron is a neural network unit (an artificial neuron) that does certain computations to detect features or business intelligence in the input data. Perceptron was introduced by Frank Rosenblatt in 1957. He proposed a Perceptron learning rule based on the original MCP neuron. A Perceptron is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time. There are two types of Perceptrons: Single layer and Multilayer. Single layer Perceptrons can learn only linearly separable patterns. Multilayer Perceptrons or feedforward neural networks with two or more layers have the greater processing power. The Perceptron algorithm learns the weights for the input signals in order to draw a linear decision boundary. This enables you to distinguish between the two linearly separable classes +1 and -1.



2.9 Convolutional Neural Network (ConvNet/CNN)

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.



A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

3.0 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications. TensorFlow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages. Developers have the option to deploy models on a number of platforms such as on servers, in the cloud, on mobile and edge devices, in browsers, and on many other JavaScript platforms. This enables developers to go from model building and training to deployment much more easily.

3.1 Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides. Keras is the most used deep learning framework among top-5 winning teams on Kaggle. Because Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster. And this is how you win. Built on top of TensorFlow 2.0, Keras is an industry-strength framework that can scale to large clusters of GPUs or an entire TPU pod. It's not only possible; it's easy.

3.2 FaceNet

FaceNet is a face recognition system developed in 2015 by researchers at Google that achieved then state-of-the-art results on a range of face recognition benchmark datasets. The FaceNet system can be used broadly thanks to multiple third-party open source implementations of the model and the availability of pre-trained models. The FaceNet system can be used to extract high-quality features from faces, called face embeddings, that can then be used to train a face identification system. FaceNet is a Deep Learning architecture consisting of convolutional layers based on GoogLeNet inspired inception models. FaceNet returns a 128 dimensional vector embedding for each face. Having been trained with triplet loss for different classes of faces (by classes I mean faces from different people) to capture the similarities and differences between

them, the 128 dimensional embedding, returned by the FaceNet model, effectively clusters faces. Hence this vector would be closer for similar faces and farther apart for dissimilar faces. This FaceNet architecture is trained over a dataset with a very large number of faces belonging to numerous classes.

| layer | size-in | size-out | kernel | param | FLPS |
|--------------|----------------------------|----------------------------|----------------------------|--------------|-------------|
| conv1 | $220 \times 220 \times 3$ | $110 \times 110 \times 64$ | $7 \times 7 \times 3, 2$ | 9K | 115M |
| pool1 | $110 \times 110 \times 64$ | $55 \times 55 \times 64$ | $3 \times 3 \times 64, 2$ | 0 | |
| rnorm1 | $55 \times 55 \times 64$ | $55 \times 55 \times 64$ | | 0 | |
| conv2a | $55 \times 55 \times 64$ | $55 \times 55 \times 64$ | $1 \times 1 \times 64, 1$ | 4K | 13M |
| conv2 | $55 \times 55 \times 64$ | $55 \times 55 \times 192$ | $3 \times 3 \times 64, 1$ | 111K | 335M |
| rnorm2 | $55 \times 55 \times 192$ | $55 \times 55 \times 192$ | | 0 | |
| pool2 | $55 \times 55 \times 192$ | $28 \times 28 \times 192$ | $3 \times 3 \times 192, 2$ | 0 | |
| conv3a | $28 \times 28 \times 192$ | $28 \times 28 \times 192$ | $1 \times 1 \times 192, 1$ | 37K | 29M |
| conv3 | $28 \times 28 \times 192$ | $28 \times 28 \times 384$ | $3 \times 3 \times 192, 1$ | 664K | 521M |
| pool3 | $28 \times 28 \times 384$ | $14 \times 14 \times 384$ | $3 \times 3 \times 384, 2$ | 0 | |
| conv4a | $14 \times 14 \times 384$ | $14 \times 14 \times 384$ | $1 \times 1 \times 384, 1$ | 148K | 29M |
| conv4 | $14 \times 14 \times 384$ | $14 \times 14 \times 256$ | $3 \times 3 \times 384, 1$ | 885K | 173M |
| conv5a | $14 \times 14 \times 256$ | $14 \times 14 \times 256$ | $1 \times 1 \times 256, 1$ | 66K | 13M |
| conv5 | $14 \times 14 \times 256$ | $14 \times 14 \times 256$ | $3 \times 3 \times 256, 1$ | 590K | 116M |
| conv6a | $14 \times 14 \times 256$ | $14 \times 14 \times 256$ | $1 \times 1 \times 256, 1$ | 66K | 13M |
| conv6 | $14 \times 14 \times 256$ | $14 \times 14 \times 256$ | $3 \times 3 \times 256, 1$ | 590K | 116M |
| pool4 | $14 \times 14 \times 256$ | $7 \times 7 \times 256$ | $3 \times 3 \times 256, 2$ | 0 | |
| concat | $7 \times 7 \times 256$ | $7 \times 7 \times 256$ | | 0 | |
| fc1 | $7 \times 7 \times 256$ | $1 \times 32 \times 128$ | maxout p=2 | 103M | 103M |
| fc2 | $1 \times 32 \times 128$ | $1 \times 32 \times 128$ | maxout p=2 | 34M | 34M |
| fc7128 | $1 \times 32 \times 128$ | $1 \times 1 \times 128$ | | 524K | 0.5M |
| L2 | $1 \times 1 \times 128$ | $1 \times 1 \times 128$ | | 0 | |
| total | | | | 140M | 1.6B |

CHAPTER 3 : REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

The client needs a smart surveillance system which can recognize the employees of their company and ensure their safety by detecting unauthorized users. The client need a system which is monitoring the employees of my firm and their various activities. The employees should be recognized and the attendance should be automatically recorded, which will give them some monthly visualizations. They need a system which can identify unauthorized users who enter my firm. They need a system that can classify the users as trusted or a threat. If the user is a threat the security of their firm needs to be deployed to that particular location.

3.2 Requirement Specification

3.2.1 Stable Internet connection

There should be a stable internet connection as the application requires internet to load the data

3.2.2 Software Quality Attributes

3.2.2.1 Reliability

The system working must not be affected by any kind of failure.

3.2.2.1 Availability

The web application will be available for the admin who want to access the his modules and work on it .

3.3 Planning and Scheduling

Planning and scheduling is an essential and complicated task of the project. Planning can be regarded as identifying all the small tasks that must be carried out in order to accomplish the goal. This plan has to be executed in the right time to complete the development of the application. Some of the major milestone involved as part of the planning is described below:

| Sl.no | Milestone Name | Milestone Description | Timeline | Remarks |
|-------|----------------------------|--|----------|---|
| 1 | Requirement Specification | All the functional and nonfunctional requirements must be collected. | 2 Weeks | Attempt should be made to add some more relevant functionalities other than those that are listed |
| 2 | Technology Familiarization | Understanding of the Technology needed to implement the project . | 2 Weeks | Understanding the technology is very important to know how fast you can develop a system, what is the best strategy to be used, how can the application developed in a faster rate, etc |
| 3 | Database creation | Listing out the attributes that should be present in the database as per user requirements. | 1 Week | It is important to finalize on the database at this stage itself so that the development & testing can proceed with the actual database itself . |
| 4 | System Design | High level view of the system needs to be written or drawn using some use-cases ,sequence diagrams etc | 2 Weeks | The design must match the intended system to be developed |

| | | | | |
|---|---|---|---------|---|
| 5 | Implementation of the front end | Implementation of the user interface and main screens | 2 weeks | During this milestone period it would be a good idea for the developer to start working on a test plan of entire system. This test plan can be updated as and when new scenarios come to mind. |
| 6 | Integrating the front end with database | The front end developed should be functional at this stage | 2 Weeks | During this period the program module connections are made with the respective databases. |
| 7 | Integration Testing | The System should be thoroughly tested by running all test cases written for the system. | 2 Weeks | During this period the system is been tested and the issues are been handled. |
| 8 | Final Review | The issues found during the previous milestones are fixed and the system is ready for the final review. | 2 Weeks | During the final review of the project, it is made sure that the application runs successfully, if the requirements are met. Also, code was reviewed so that, the code can be reused in the future. |

3.4 Software and Hard Requirements

Hardware Requirement :

RAM – 8 GB minimum
Nvidia 1050 GTX minimum

Software Requirement :

BackEnd Language – Python
FrontEnd Language – Flask-python, Dart
Neural Network Requirements – CUDA 9.0 minimum, tensorflow-gpu
Database – Mongo DB
IDE – Visual Studio Code, Atom

Performance Requirements :

The performance shall depend upon hardware components of the user. The system needs to be in a tensorflow environment with minimum of Nvidia 1050 GTX GPUs and 8gb of RAM. The system needs have software components that has to run the neural network like CUDA, tensorflow-gpu and Keras. The system also needs a high quality camera.

Safety Requirements :

For the applications to work efficiently it's recommended that no manual intervention to the database should happen. The information of the users and data should be safe and only authorized users should be able to access the information and manipulate the data on the web app. All the data is stored safely and is maintained securely on the servers.

Security Requirements :

User authentication is done by authorization object which helps in restricting data access to an unauthorized person. The information of the employees will be safe and only authorized users can access the backend system to modify the permissions of the app users. There is no access to the login

Communications Interfaces :

The system requires the following communication interfaces to operate at full functionality.

- Network connection to communicate between a client and server.
- Web browser to interact with the system.

3.5 Preliminary Product Description

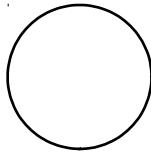
This project aims to enhance the physical security of an enterprise which will help the security team for quick real-time decisions. It also wants to improve the facial attendance system to classify the employees more accurately

3.6 Conceptual Models

3.6.1 Data Flow Diagrams

Data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system; it differs from the flowchart as it shows the data flow instead of the control flow of the program.

A data flow diagram can also be used for the visualization of data processing. DFDs show the flow data from external entities into the system, showed how the data moved from one process to another, as well as its logical storage. The basic notation used to create a DFD makes it easy to analyze and understand.

| SYMBOL | NOTATION | DESCRIPTION |
|---|--------------------|--|
|  | Source/Destination | Sources and destinations (sink) define the system's boundaries. It is represented by a rectangle. |
|  | Process | Represents an activity that changes, moves or otherwise transforms data. It is shown as a round-cornered rectangle/oval. |
|  | Data store | Represents data at rest and implies that |

| | | |
|---|-----------------------------|--|
| | | the data are held between processes. It is shown as an open-ended, horizontal rectangle. |
|  | Flow of activity or control | Represents data in motion. It is depicted with an arrow. |

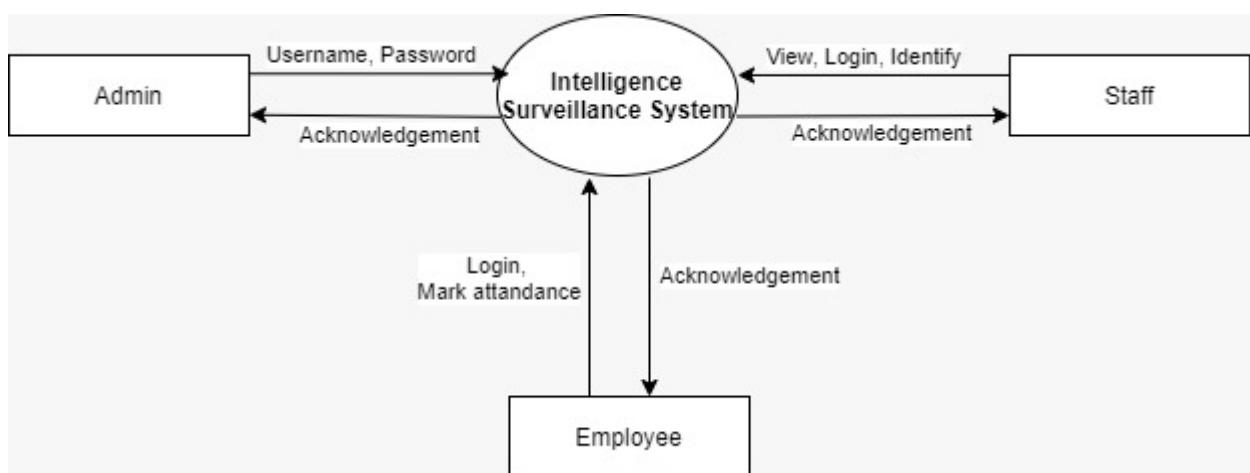


Figure 1 .Zero level DFD for Face Recognition and Attendance System

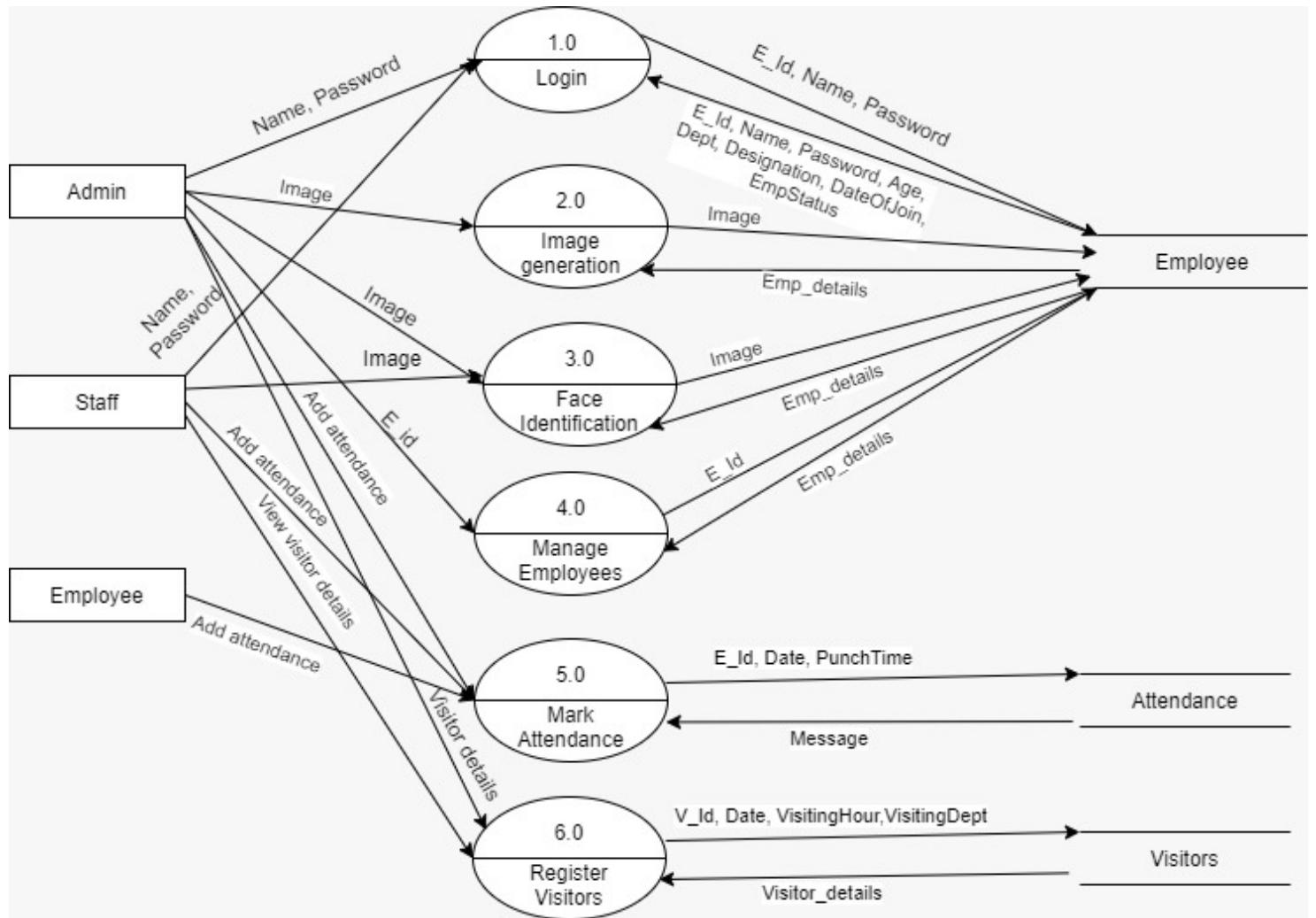


Figure 2. Level 1 DFD for Face Recognition and Attendance System

CHAPTER 4 : SYSTEM DESIGN

4.1. Basic Modules

The proposed project has following modules :

4.1.1 Face Recognition Module

4.1.2 Attendance Module

4.1.1 Face Recognition Module :

Staff can scan the face of any person to get his details. The scanned face will be sent to the neural network which will classify the face. Once the face is classified the person's details are retrieved from the database and displayed on the screen. If the person's face is classified as an unauthorized personal then security will be alerted.

4.1.2 Attendance Module :

The person can scan his face to punch in the attendance. The system will scan the face and send it to the neural network which will in turn classify the face. If the face gets classified the attendance will be punched in. If the face returns as an unauthorized person the security staff will be alerted.

4.2 Data Design Project Structure

4.2.1. Schema Design

Table 1. employee_table

| Column Name | Data Type |
|-----------------|------------|
| E_Id | Text(10) |
| Name | Text(30) |
| Age | Numeric(2) |
| Password | Text(12) |
| Designation | Text(20) |
| Department | Text(20) |
| Date of Joining | Date |
| EmpStatus | Text(10) |

Table 2. attendance_table

| Column Name | Data Type |
|-------------|-----------|
| E_Id | Text(10) |
| Date | Date |
| PunchTime | time |

Table 3. visitor_table

| Column Name | Data Type |
|---------------------|------------------|
| V_Id | Text(10) |
| Date | Date |
| Visiting_Department | Text(20) |
| Visiting_Hours | time |

4.3. Procedural Design

4.3.1. Logic Diagrams

4.3.1.1 Sequence Diagrams

A Sequence Diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes.

- User or another system that will interact with the objects. Actors are placed in columns.
- The objects in the system that is being modelled. Represented in rectangular boxes.
- Dashed line is lifeline which indicates the existence of the object over time. Rectangular box is Activation which indicates that the object is performing an action.

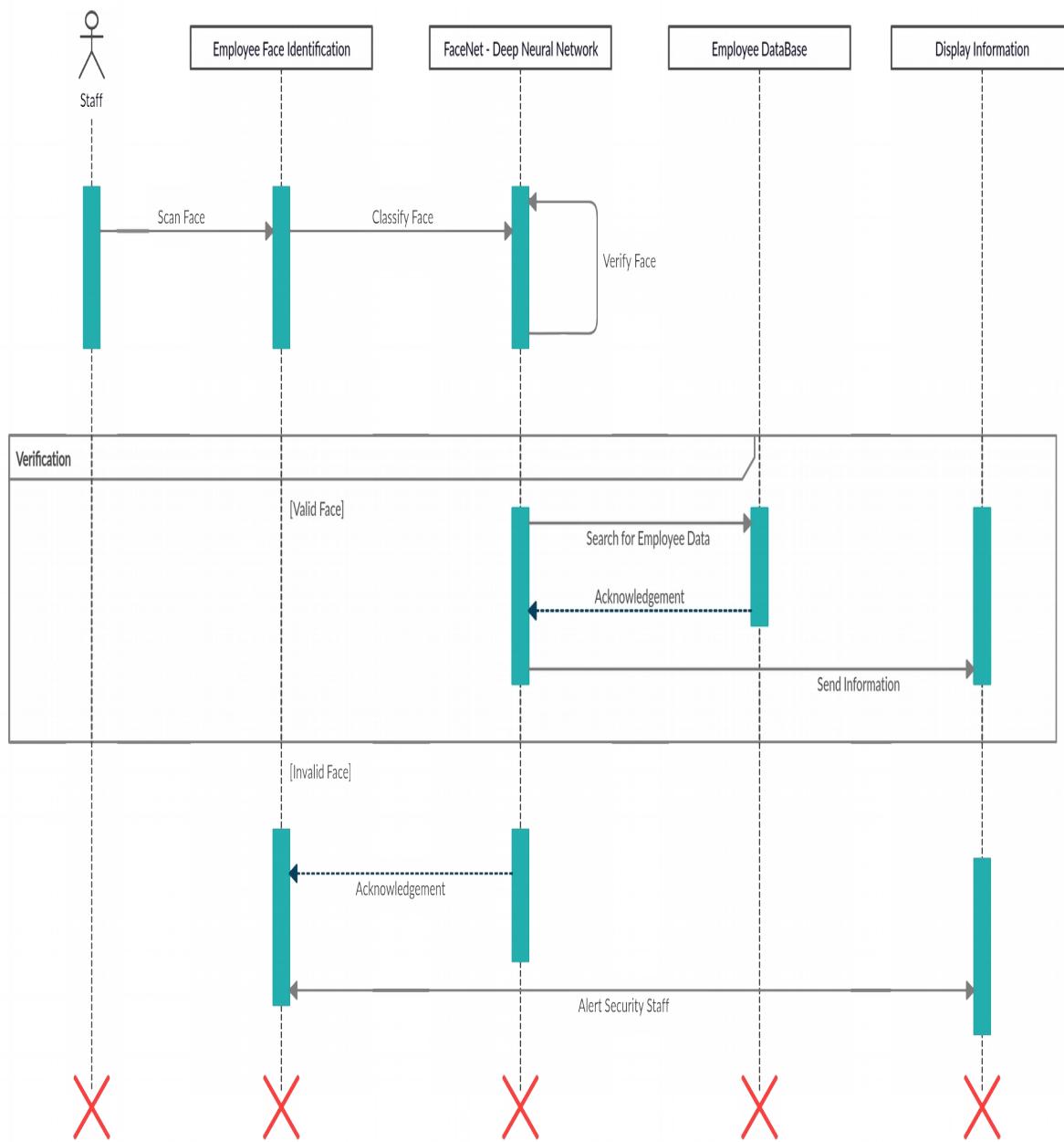


Figure 3. Sequence Diagram for Face Recognition

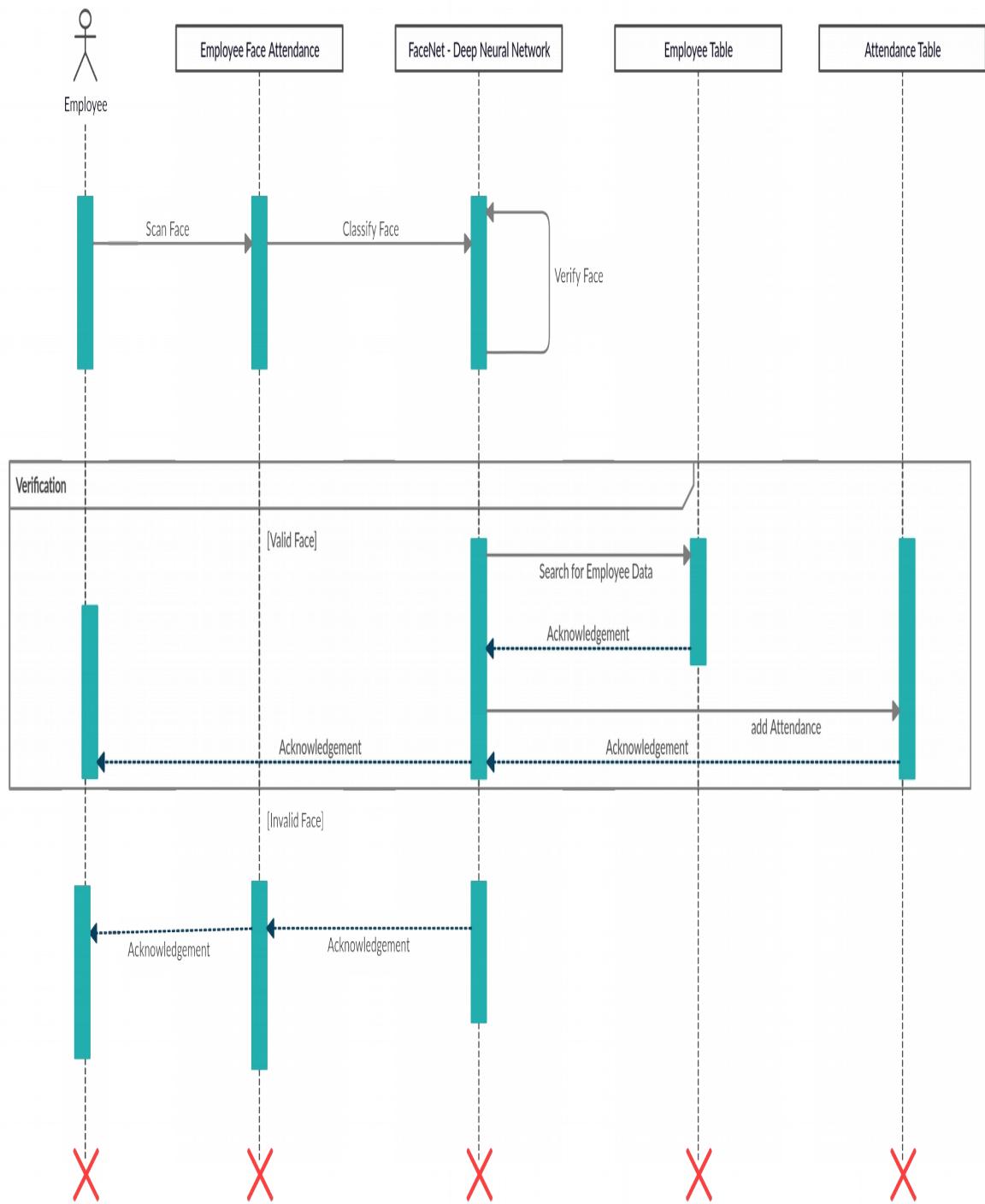
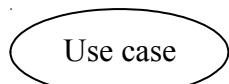


Figure 4. Sequence Diagram for Attendance System

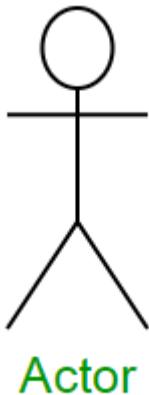
4.3.1.2. Use Case Diagrams

Use Case Diagram Symbols and Notations. Use case diagrams model the functionality of system using actors and use cases. Use case diagrams depict:

- **System:** Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.
- **Use Case:** Draw use cases using ovals. Label the ovals with verbs that represent the system's functions. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.



- **Actors:** Actors are the users of a system. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures. When one system is the actor of another system, label the actor system with the actor stereotype.



- **Scenario:** A scenario is one hypothetical instance of how a particular use case might play out. A single use case thus inspires many different scenarios, in the same way that planning a driving trip from one city to another can involve many different routes.
- **Associations:** Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

- **System boundary boxes (optional):** You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system.

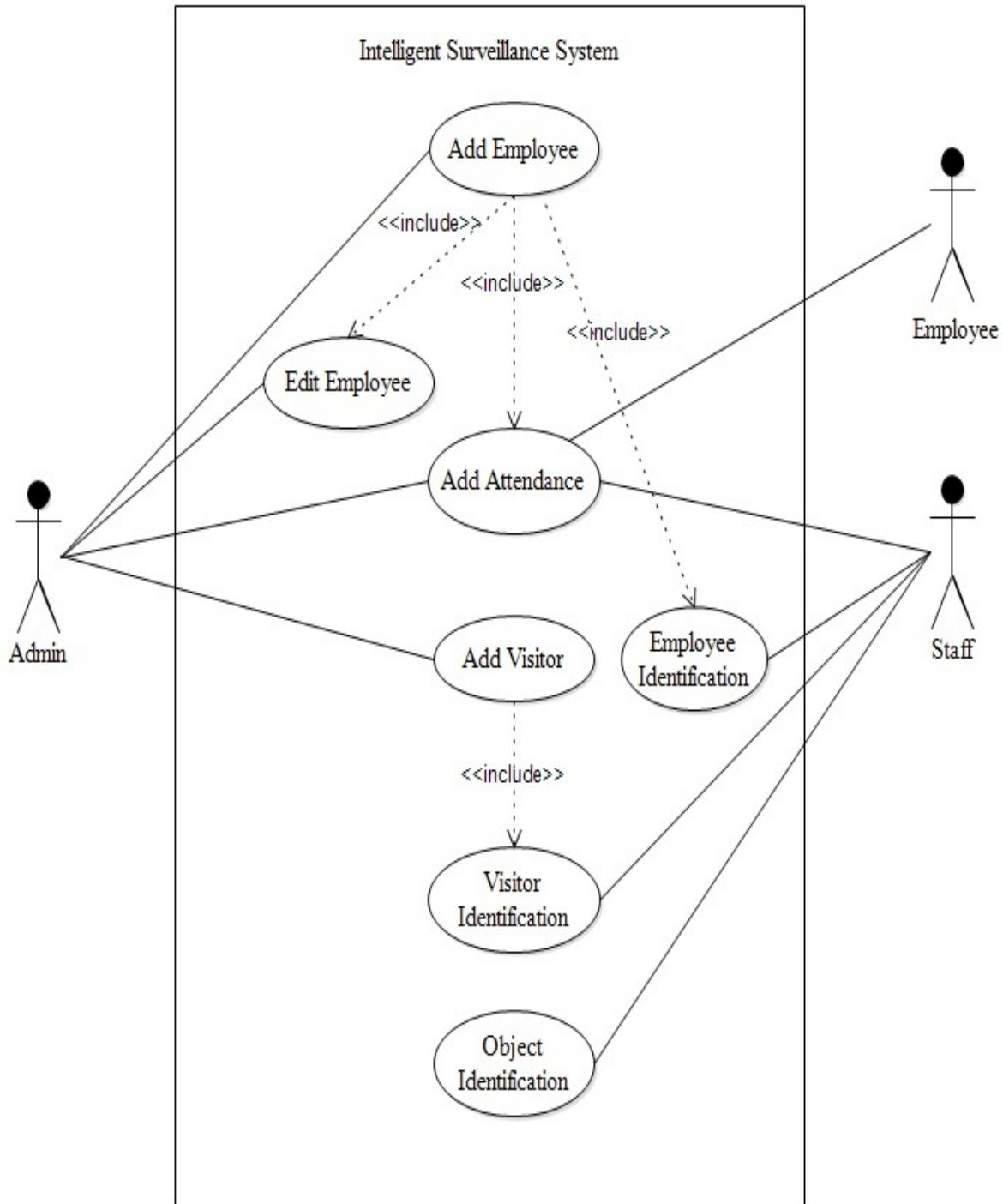
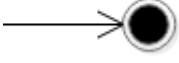
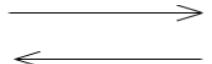


Figure 5. Use case Diagram for Intelligent Security System

4.3.1.3 Activity Diagrams

Activity diagrams are used to describe the business and operational step-by-step workflows of components in a system. Rounded rectangles represent activities, Diamonds represent decisions, a black circle represents the start (initial state) of the work flow, and an encircled black circle represents the end (final state). Arrows run from the start towards the end and represent the order in which activities happen.

| Shape | Notation | Description |
|---|-----------------------------|--|
|  | Initial Activity | Represents the starting point or first activity of the flow. It is denoted by a solid circle. |
|  | Final Activity | Represents the end of activity diagram, also called as final activity. It is shown by a bull's eye symbol. |
|  | Activity | Represents an activity. It is shown by a rectangle with rounded (almost oval) edges. |
|  | Flow of Activity or Control | Shows the direction of the workflow in the activity diagram. It is depicted with an arrow. |

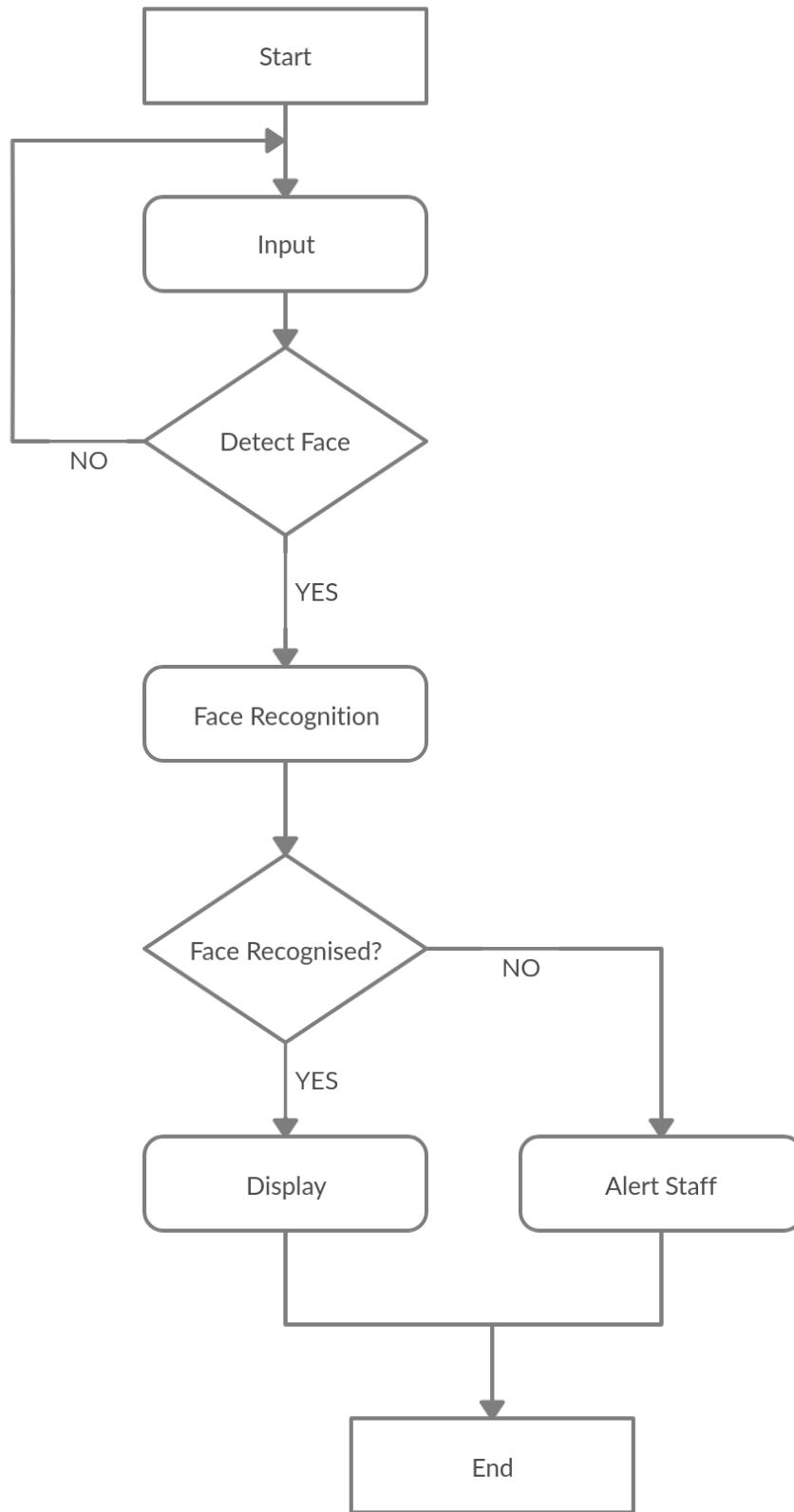


Figure 6. Activity Diagram for Face Recognition

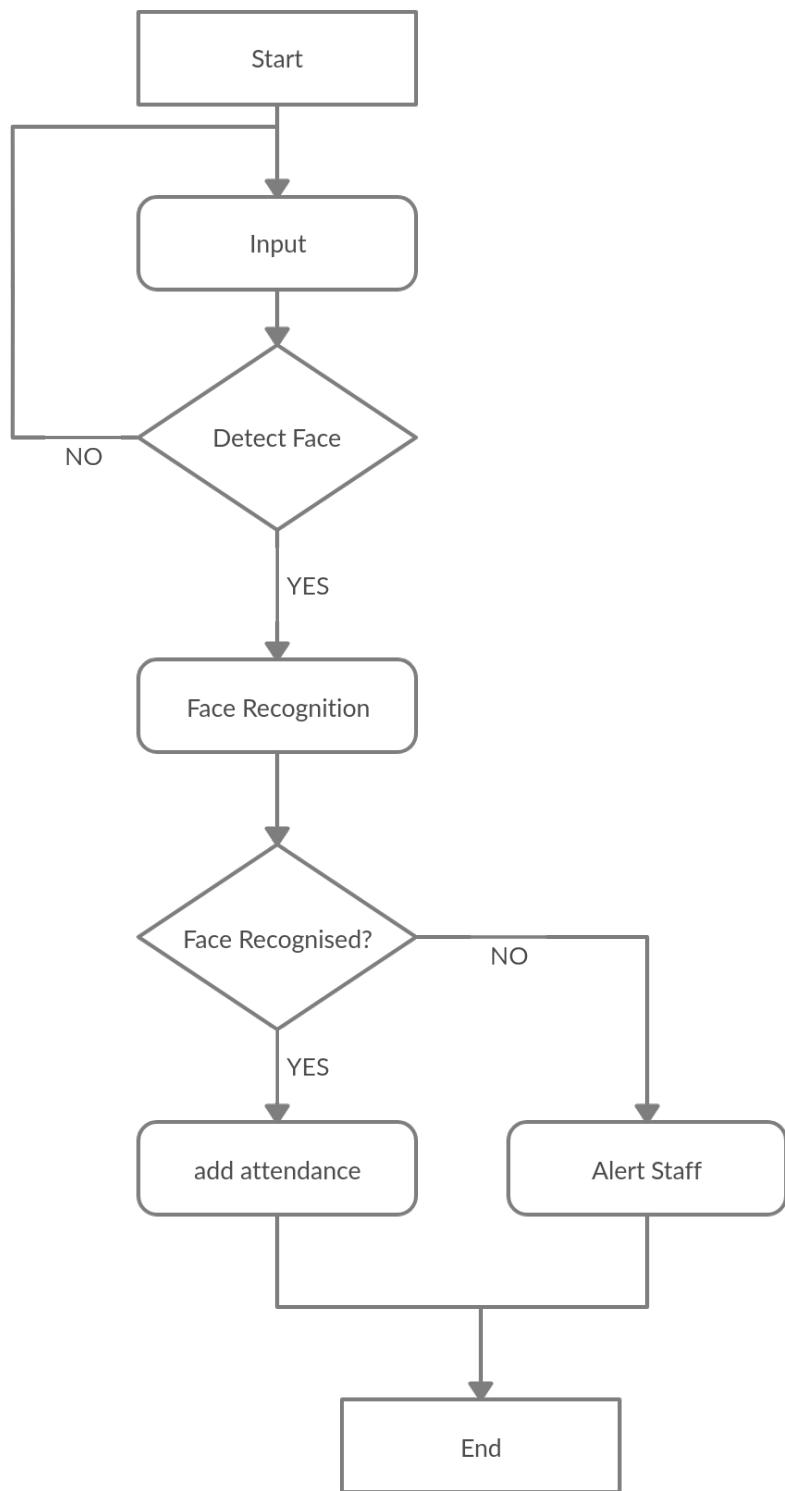


Figure 7. Activity Diagram for Attendance System

4.3.1.4 State chart Diagrams

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system.

- **States**

State represent situations during the life of an object. You can easily illustrate a state in Smart Draw by using a rectangle with rounded corners.

- **Transition**

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself.

- **Initial State**

A filled circle followed by an arrow represents the object's initial state.

- **Final State**

An arrow pointing to a filled circle nested inside another circle represents the object's final state.

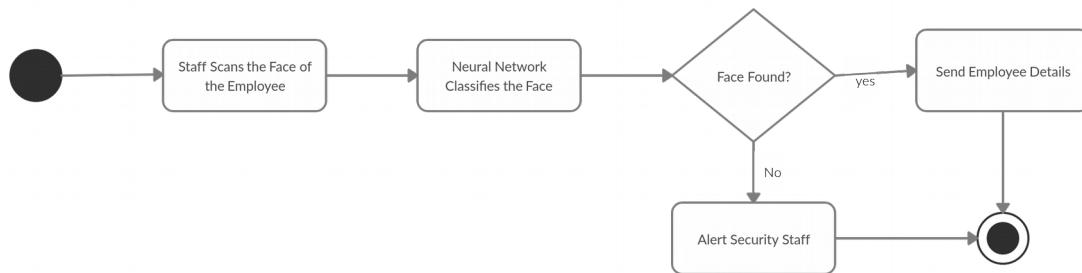


Figure 8. State Chart Diagram for Face Recognition

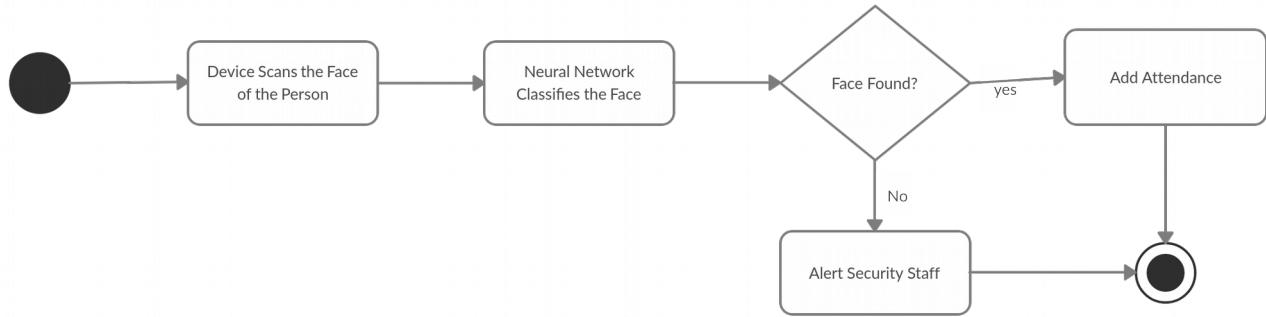


Figure 9. State Chart Diagram for Attendance

4.3.1.5 Class Diagrams

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row contains the attributes of the class, and the bottom section expresses the methods or operations that the class may use. Classes and subclasses are grouped together to show the static relationship between each object.

Basic Class Diagram Symbols and Notations

- **Classes**

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes. Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered,

bolded, and capitalized), list the attributes in the second partition (leftaligned, not bolded, and lowercase), and write operations into the third.

- **Active Classes**

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

- **Visibility**

Use visibility markers to signify who can access the information contained within a class. Private visibility, denoted with a - sign, hides information from anything outside the class partition. Public visibility, denoted with a + sign, allows all other classes to view the marked information. Protected visibility, denoted with a # sign, allows child classes to access information they inherited from a parent class.

- **Associations**

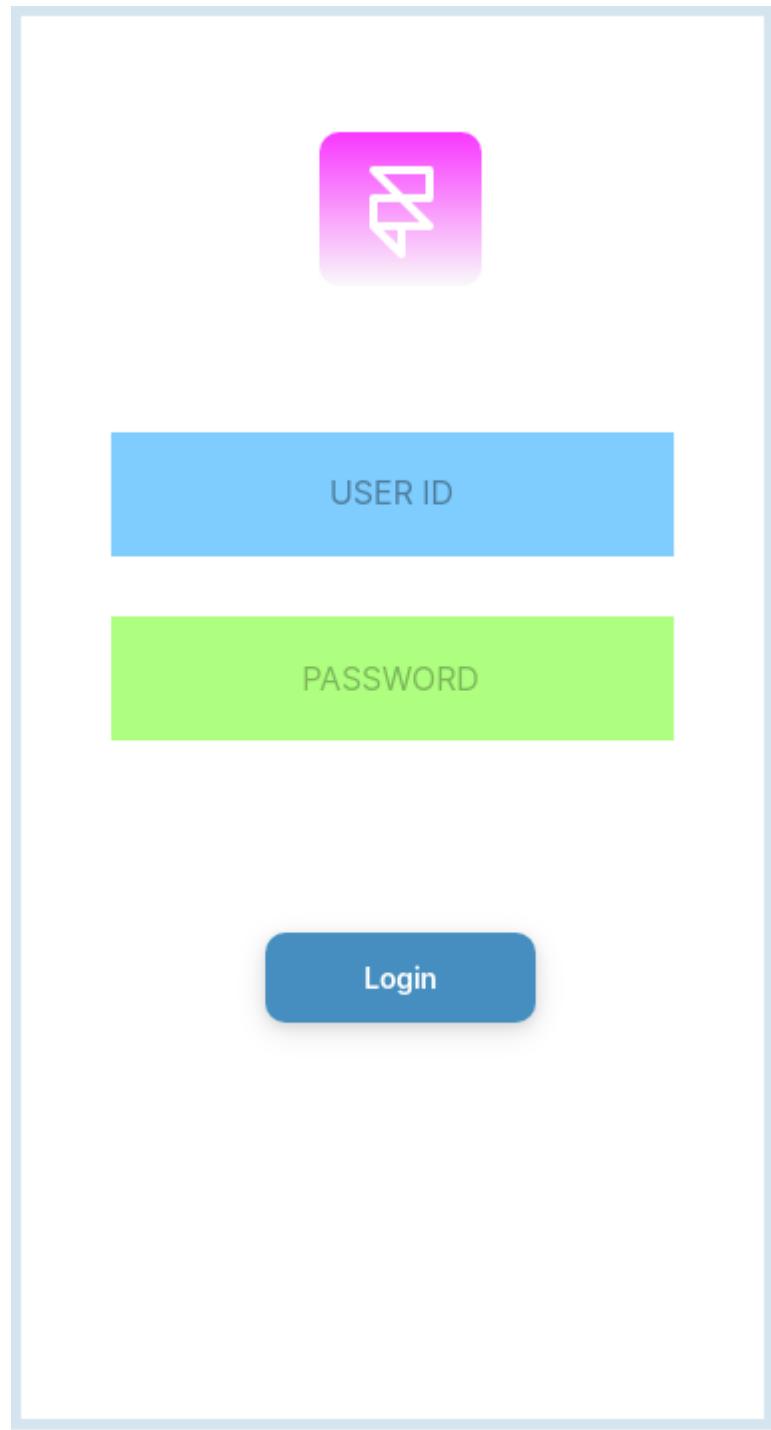
Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each

- **Multiplicity (Cardinality)**

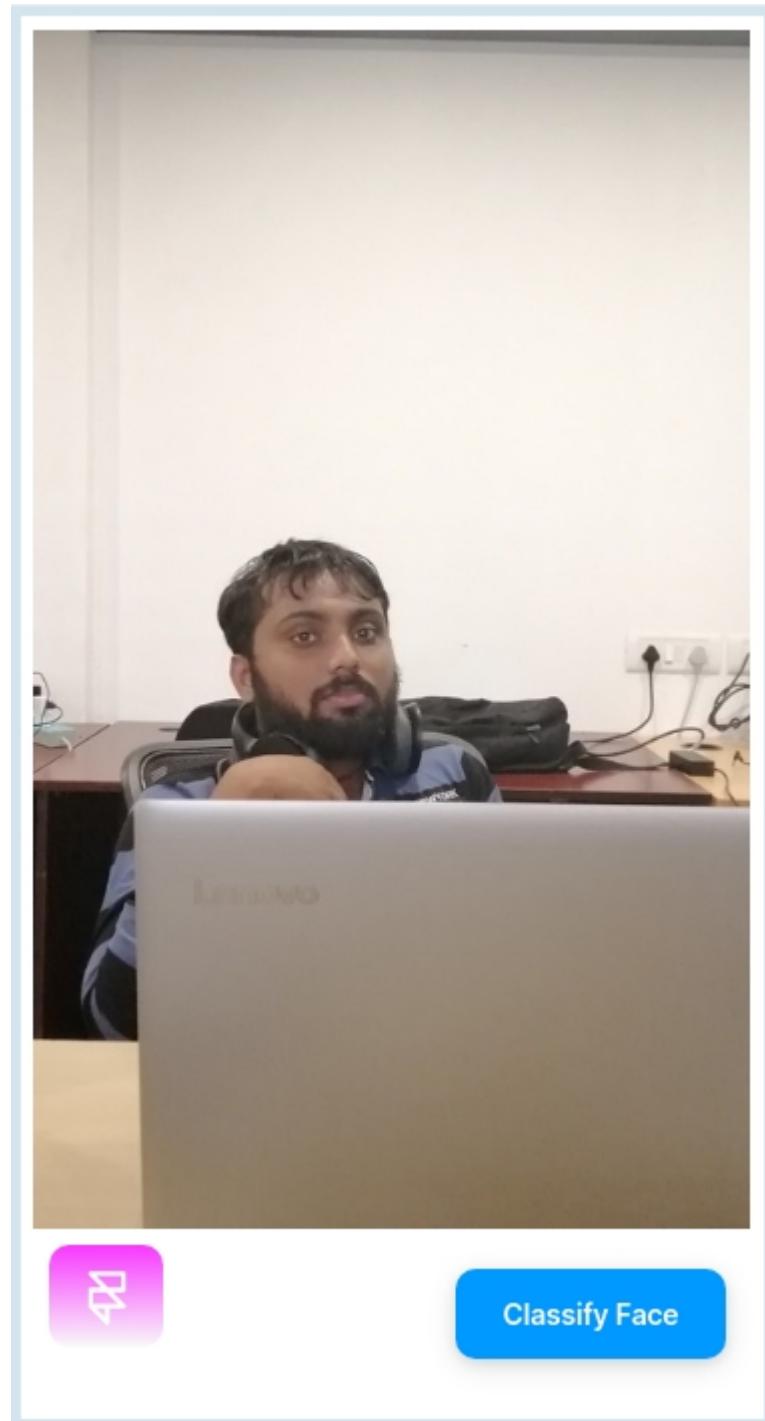
Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for just one company.

4.4 User interface Design

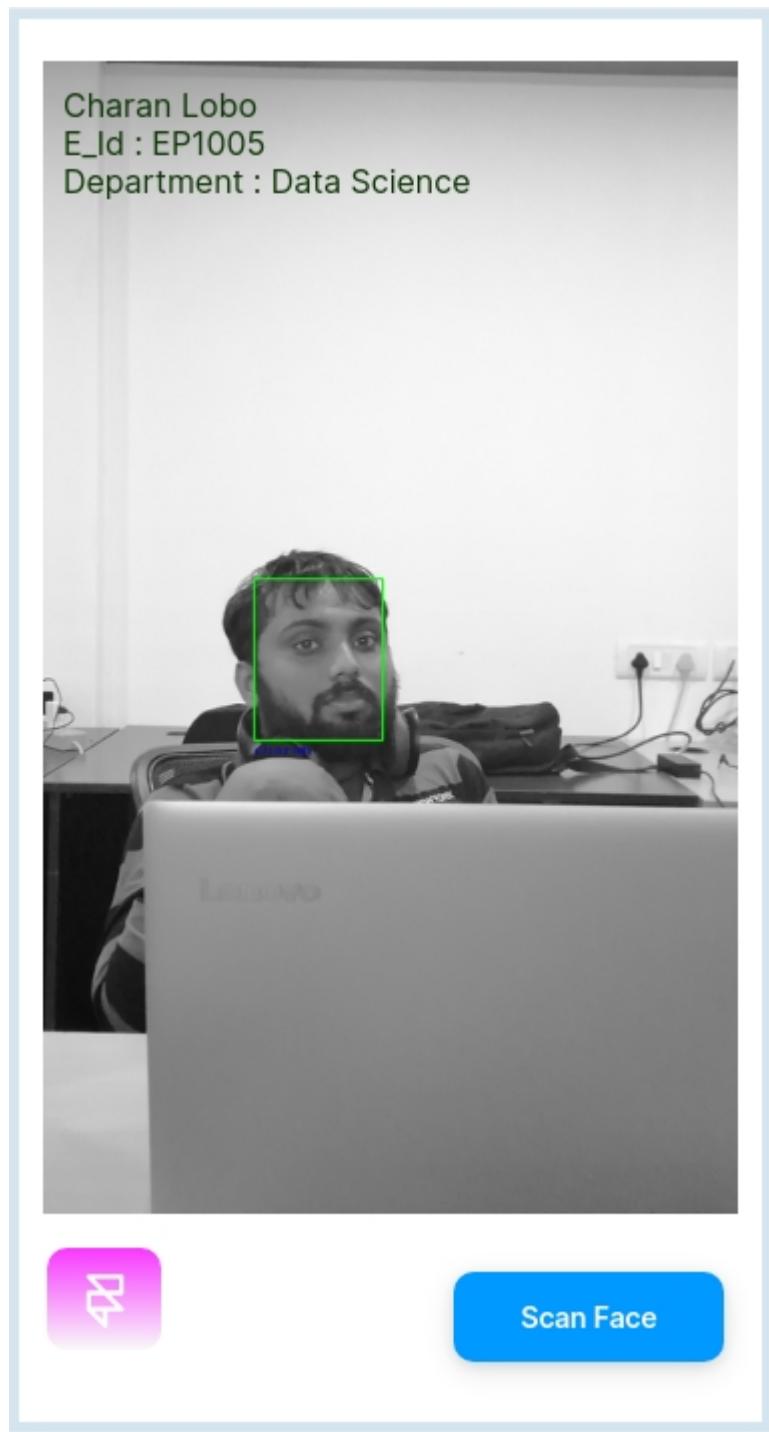
4.4.1 Login Screen



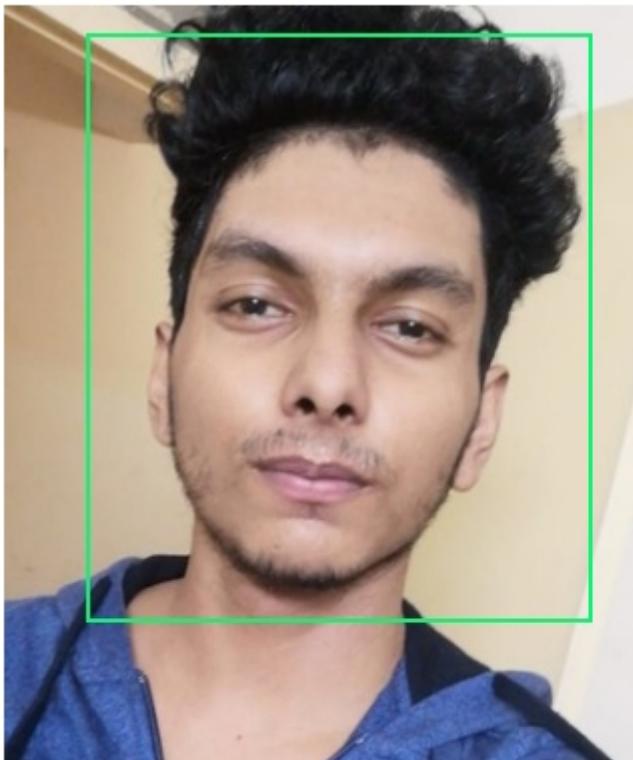
4.4.2 Face Scan Screen



4.4.3 Face Recognition



4.4.3 Attendance System



Align your Face Properly



Attendance is Punched in.
Proceed.

developed by ihub IT Solutions

4.5 Security Issues

If the Staff shares the login info with others it might compromise the system. If the Admin Account gets accessed it will pose a threat to the neural network

4.6 Test Case Design

▪ Software testing

It is a process used to identify the correctness, completeness and quality of developed computer software i.e., software testing is to execute a software program with the intent of finding bugs.

▪ Software quality assurance

It involves the entire software development process monitoring and improving the process, making sure that any agreed upon standards and procedures are followed and ensuring that problems are found and dealt with. It is oriented to ‘Prevention’.

▪ SOFTWARE DEVELOPMENT LIFE CYCLE

- Requirements
- Analysis
- Design
- Coding
- Testing
- Maintenance

▪ APPROACHES TO TESTING

There are two basic approaches to testing:

1. Black-Box Testing
2. White-Box Testing

1) Black-Box Testing

In black box testing the structure of the program is not considered. Test cases are decided solely on the requirements or specification of the program or module, and the internals of the module or the program are not considered for selection of test cases. In black-box testing, the tester only knows the inputs that can be given to the system and what output the system

should give. In other words, the basis for deciding test cases in functional testing is the requirements or specifications of the system or module. This form of testing is also called functional or behavioural testing.

2) White Box Testing

White box testing is concerned with the function that the tested program is supposed to perform and does not deal with the internal structure of the program responsible for actually implementing that function thus black-box testing is concerned with the functionality rather than implementation of the program. White-box testing, on the other hand is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions (although that may be a by-product) but to exercise the different programming structures and data structures used in the program. White-box testing is also called structural **testing**

CHAPTER 5 : IMPLEMENTATION AND TESTING

5.1. Implementation Approaches

Implementation is the stage in the project where theoretical design is turned into a working system and is giving confidence on the new system for the users that will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users.

The objective of the system implementation phase is to implement a fully functional system, which deliverables met in order of priority and with effective training. The end result of this phase is an accurately functioning system with properly trained users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself. During the implementation phase, the system is built according to the specifications from the previous phases. This includes writing code, performing code reviews, performing tests, selecting components for integration, configuration, and integration.

5.2. Coding Details and Code Efficiency

To keep the code standard and efficient so that the application is lightweight and robust the following standards as defined by ORACLE were followed. The following were eliminated using the inbuilt code efficiency tools provided by the IDE and also by subsequent code reviews

- Unused local variables
- Empty catch blocks
- Unused parameters
- Empty 'if' statements
- Unused private methods
- Short/long variable and method names

Moreover, to check the efficiency of the coding methods used Code Coverage tools provided with the IDE was used to check the following branch coverage, loop coverage and strict condition coverage (decision coverage)

5.3 Testing Approach

Testing is the important step of s/w development. An elaborate testing of the data is prepared and the system is using the test data. While testing, errors are noted and correction is made. The users are trained to operate the developed system. The product is tested using unit testing, integration testing and system testing. Unit testing is an approach to software development in which tests are written for each functionality in the application. This is performed as and when the modules are developed. Each module is thoroughly tested and integrated upon which integration testing will be done to test whether the modules communicate with each other in the expected manner. It is done to weed out errors and bugs caused due to the integration. Finally, System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of Black box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is aimed at ensuring the system works accurately before the live operation commences. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. A series of testing are performed for the proposed system before the system is ready for user acceptance testing.

Characteristics of a Good Test:

- Tests are likely to catch bugs
- No redundancy
- Not too simple or too complex

5.3.1 Unit Testing

Unit testing focuses on testing the smallest unit of the module. To check whether each module in the software works properly so that it gives desired outputs to the given inputs. All validations and conditions are tested in the module level in the unit test. Control paths are tested to ensure the information properly flows into, and output of the program unit and out of the program unit under test. Boundary conditions are tested to ensure that the modules operate at boundaries.

5.3.2 Integration Testing

The major concerns of integration testing are developing an incremental strategy that will limit the complexity of entire actions among components as they are added to the system. Developing a component as they are added to the system, developing an implementation and integration schedules that will make the modules available when needed, and designing test cases that will demonstrate the viability of the evolving system.

5.3.3 User Acceptance Testing

This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

5.3.4 Performance Testing

It is mostly used to identify any bottlenecks or performance issues rather than finding bugs in a software.

5.3.5 Load Testing/Stress Testing

It is a process of testing the behaviour of a software by applying maximum load in terms of software accessing and manipulating large input data. It can be done at both normal and peak load conditions. This type of testing identifies the maximum capacity of software and its behaviour at peak time. Stress testing includes testing the behaviour of a software under abnormal conditions. For example, it may include taking away some resources or applying a load beyond the actual load limit.

5.4 Modifications and Improvements

The bugs & defects identified at each of the stages were fixed as and when they are found. The regression testing is not performed completely as the product is not yet reached the released stage. Some enhancements have been given by the customer when they saw a demo of the product. Proper measures are taken to incorporate these enhancements in the product.

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

User Login

| SL.N | Test Case | Expected Result | Test Result |
|-------------|----------------------------------|------------------------|--------------------|
| 0 | | | |
| 1 | ID not entered | Enter ID | Successful |
| 2 | Invalid password entered | Password Incorrect | Successful |
| 3 | Invalid password entered 5 times | Account Blocked | Successful |

Face Recognition

| SL.N | Test Case | Expected Result | Test Result |
|-------------|---------------------|------------------------|--------------------|
| 0 | | | |
| 1 | Face not on screen | Face not detected | Successful |
| 2 | Face not recognized | unauthorized | Successful |

6.2 User Documentation

The application is not very complicated for the customer to comprehend. It is has easy navigation process wherein the user can smoothly navigate through the process and complete the order effectively , so it won't be necessary to provide any user documentation.

Z

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

The "Face Recognition and Attendance System " is an easy way to manage the physical security of the firm. The Security Staff don't need wait for a lot of time to apprehend a suspect. They can find out if the suspect is unauthorized in a matter of seconds. The Attendance system is an upgrade over all the other face recognition system out there as it uses deep learning which has a better classification rate than all the other methods

7.2 Limitations of the System

- Internet connection is required
- Need a very powerful system to run the neural network

7.3 Future Scope of the Project

- This system can be turned into a IOT based system which can detect and identify persons from the camera footage

CHAPTER 8: REFERENCES

Books Referenced

- An Integrated Approach to Software Engineering – Pankaj Jalote.
- Software Engineering, 5th Edition, Addison- Wesley – Ian Somerville.
- Elements of software Project Development.
- Unified Modelling Language User Guide SECOND EDITION – Grady Booch
- Software Engineering by Ian Sommerville
- Neural Networks and Learning Machines by Simon Haykin

Website Referred

- <http://www.iiitd.edu.in/~jalote/jalotesebook>
- http://en.wikipedia.org/wiki/Software_Requirements_Specification
- www.medium.com
- www.towardsdatascience.com
- www.kaggle.com

CHAPTER 9: GLOSSARY

| Abbreviation | Description |
|---------------------|-----------------------------------|
| SDD | Software Design Document |
| GUI | Graphical User Interface |
| SRS | Software Requirement Specificatio |
| ERD | Entity Relationship Diagram |
| DFD | Data Flow Diagram |



**ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE - 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY**

FEEDBACK FORM for DISSERTATION-I

Student Register Number : _____

Student Name : _____

Title of the Project : _____

Name of the company : _____

Date of Dissertation - I : _____

I. Strengths of the System Requirement Specification

- 1.
- 2.
- 3.

II. Weakness of the System Requirement Specification

- 1.
- 2.
- 3.

III. Suggestions for improving the System Requirement Specification

- 1.
- 2.
- 3.
- 4.
- 5.

6.

Name & Signature
Examiner -I

Name & Signature
Examiner -II

Name & Signature
Internal Guide



**ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY**
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE - 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

FEEDBACK FORM for DISSERTATION-II

Student Register Number : _____

Student Name : _____

Title of the Project : _____

Name of the company : _____

Date of Dissertation - II : _____

I. Strengths of the System Design Document

- 1.
- 2.
- 3.

II. Weakness of the System Design Document

- 1.
- 2.
- 3.

III. Suggestions for improving the System Design Document

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Name & Signature
Examiner -I

Name & Signature
Examiner -II

Name & Signature
Internal Guide



**ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION
TECHNOLOGY**
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE - 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

FEEDBACK FORM for DISSERTATION-III

Student Register Number : _____

Student Name : _____

Title of the Project : _____

Name of the company : _____

Date of Dissertation - III : _____

I. Strengths of the Mid Term Evaluation

- 1.
- 2.
- 3.

II. Weakness found during Mid-Term Evaluation

- 1.
- 2.
- 3.

III. Suggestions for improving the System Development process

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Name & Signature
Examiner -I

Name & Signature
Examiner -II

Name & Signature
Internal Guide