

A Project Report

on

Fraud Detection at Shopping Malls using Facial Recognition

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

SYED AMAIR ALI

(160316733019)

SHAIK ABDUL FARAZ

(160316733024)

MOHD SOHAIL AHMED

(160316733031)

Under the guidance

of

Mrs. Neha Unissa

Assistant Professor

Department Of CSE, DCET



Department of Computer Science Engineering

Deccan College of Engineering and Technology

(Affiliated to Osmania University)

Hyderabad

2019 - 2020

CERTIFICATE

This is to certify that the project report entitled **Fraud Detection at Shopping Malls using Facial Recognition** being submitted by **Mr. Syed Amair Ali (160316733019)**, **Shaik Abdul Faraz (160316733024)** and **Mr. Mohd Sohail Ahmed (160316733031)** in partial fulfilment for the award of the Degree of Bachelor of Engineering in Computer Science by the Osmania University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Internal Guide

Mrs. Neha Unissa

Asst. Professor

Department of CSE

DCET, Hyderabad

External Examiner

Head of the Department

Dr. Syed Raziuddin

Professor

Department of CSE

DCET, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled “**Fraud Detection at Shopping Malls using Facial Recognition**” is a record of work done by us in the Department of Computer Science & Engineering, Deccan College of Engineering and Technology, Osmania University, Hyderabad in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Computer Science & Engineering.

The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in this dissertation have not been submitted to any other university for the award of any degree or diploma.

SYED AMAIR ALI

(160316733019)

SHAIK ABDUL FARAZ

(160316733024)

MOHD SOHAIL AHMED

(160316733031)

ACKNOWLEDGEMENT

We thank the almighty for giving us the courage and perseverance in completing the main-project. This project itself is acknowledgements for all those people who have given us their heartfelt co-operation in making this project a grand success.

We are thankful to our principal Dr. M.A. MALIK, for providing the necessary infrastructure and labs and also permitting to carry out this project. With extreme jubilation and deepest gratitude, We would like to thank Head of the C.S.E. Department, Dr. Syed Raziuddin, for his constant encouragement.

We are greatly indebted to project guide Ms. Neha Unissa, Assistant Professor, Computer Science and Engineering, for providing valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by her.

We are thankful to our lab faculty for providing the necessary Infrastructure and labs and also permitting to carry out this project.

Our special thanks to all the faculty of Computer Science and Engineering and peers for their valuable advises at every stage of this work. Last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

ABSTRACT

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Fraud Detection at Shopping Malls using Facial Recognition is a real-world software solution which comes with the day-to-day activities that allows the shopping mall's owner to track the person, if he has done some sort of damaged to the goods or property of the owner or steal some goods and left behind without getting caught.

Fraud Detection at Shopping Malls using Facial Recognition, is a process of recognizing the known faces or the fraud persons face from the live video stream. In our project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera at the entrance probably. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Deep Learning using Transfer Learning. It helps in conversion of the frames of the video into images so that the face of the fraud persons can be easily recognized so that a notification can be sent to the all employees which will run our android application lately.

Table of Contents

I.	Project Description.....	14
1)	Project Overview.....	14
2)	The Purpose of the Project	15
2.1)	The User Business or Background of the Project Effort	15
2.2)	Goals of the Project	15
2.3)	Measurement	15
3)	The Scope of the Work.....	16
3.1)	The Current Situation	16
3.2)	The Context of the Work	16
3.3)	Work Partitioning	17
3.4)	Competing Products	19
4)	Stakeholders	19
4.1)	The Client	19
4.2)	The Customer	19
4.3)	Hands-On Users of the Product	19
4.4)	Priorities Assigned to Users	20
4.5)	Maintenance Users and Service Technicians	20
5)	Mandated Constraints.....	20
5.1)	Solution Constraints	20
5.2)	Implementation Environment of the Current System.....	20
5.3)	Partner or Collaborative Applications	20
5.4)	Off-the-Shelf Software	20
5.5)	Schedule Constraints	21
5.6)	Budget Constraints	21
6)	Naming Conventions and Definitions	21
7)	Relevant Facts and Assumptions.....	22
II.	Project Description.....	23
8)	Relevant Facts and Assumptions.....	23
8.1)	Use Case Diagrams.....	23
8.2)	Product Use Case List.....	24
8.3)	Individual Product Use Cases.....	25
8.3.1)	Specification of actors.....	25
8.3.1.1)	Employee:	25
8.3.1.2)	System Admin:	25

8.3.1.3)	Worker/Workers:.....	26
8.3.2.1)	Admin Log in Use Case:.....	26
8.3.2.2)	Register of Employee Use Case:	27
8.3.2.3)	Deletion of Employee Use Case:.....	27
8.3.2.4)	Modification of Employee's Data Use Case:.....	28
8.3.2.5)	Employee Log-In Use Case:.....	28
8.3.2.6)	Employee Password Generator Use Case:.....	29
8.3.2.7)	Employee Upload Data Use Case:.....	29
8.3.2.8)	Employee Delete/Modify Data Use Case:.....	30
9)	Functional Requirements.....	31
9.1)	Summary of Functions.....	31
9.2)	Functional Requirements	31
1)	Authentication	31
2)	User Accounts	32
3)	Password Generator	32
4)	Data Accessibility	32
5)	Entity Profile Administration.....	32
6)	Data Integrity	32
7)	System Security	32
9.3)	Use Cases.....	33
9.3.1)	Employee Use Case	33
9.3.2)	Admin Use Case	35
10)	Data Requirements	36
10.1)	Data Description	36
10.2)	Data Objects:.....	36
10.3)	Relationships.....	37
11)	Performance Requirements	38
11.1)	Speed and Latency Requirements	38
11.2)	Precision or Accuracy Requirements	38
11.3)	Capacity Requirements	38
12)	Dependability Requirements	39
12.1)	Reliability Requirements	39
12.2)	Availability Requirements	39
12.3)	Robustness of Fault-Tolerance Requirements	39
12.4)	Safety-Critical Requirements.....	40

13)	Maintainability and Supportability Requirements	40
13.1)	Maintenance Requirements.....	40
13.2)	Supportability Requirements	40
13.3)	Adaptability Requirements	41
13.4)	Scalability or Extensibility Requirements	41
13.5)	Longevity Requirements	41
14)	Security Requirements	42
14.1)	Access Requirements	42
14.2)	Integrity Requirements.....	42
14.3)	Privacy Requirements	42
14.4)	Immunity Requirements.....	42
15)	Usability and Humanity Requirements	43
15.1)	Ease of Use Requirements	43
15.2)	Personalization and Internationalization Requirements.....	43
15.3)	Learning Requirements	43
15.4)	Understandability and Politeness Requirements.....	43
15.5)	Accessibility Requirements	43
15.6)	User Documentation Requirements	44
15.7)	Training Requirements.....	44
16)	Look and Feel Requirements.....	44
16.1)	Appearance Requirements	44
16.2)	Style Requirements	44
17)	Operational and Environmental Requirements	45
17.1)	Expected Physical Environment	45
17.2)	Requirements for Interfacing with Adjacent Systems	45
17.3)	Producing Requirements	45
17.4)	Release Requirements	45
18)	Cultural and Political Requirements	46
18.1)	Cultural Requirements	46
18.2)	Political Requirements	46
19)	Legal Requirements.....	46
19.1)	Compliance Requirements	46
III.	Design	47
20)	System Design.....	47
20.1)	Design Goals	47

20.1.1)	From the Perspective of Users	47
20.2)	From the Perspective of us.....	47
21)	Proposed Software Architecture.....	48
21.1)	Overview	49
21.2)	Class Diagram.....	50
21.3)	Dynamic Model	51
21.3.1)	Activity Diagram.....	51
21.3.2)	Sequence Diagram	52
21.4)	Subsystem Decomposition.....	53
21.5)	Hardware / Software Mapping.....	54
21.6)	Data Dictionary	55
21.6.1)	Fraud Person Data.....	55
21.6.2)	Employee.....	55
21.6.3)	Admin.....	55
21.7)	Persistent Data Management.....	56
21.8)	Access Control and Security	57
22)	Implementation.....	58
23)	Screenshots.....	66
IV.	Test Plans	79
24)	Features to be Tested / Not to be Tested	79
a)	Features to be Tested	79
b)	Features Not to be Tested	79
25)	Pass / Fail Criteria	80
I)	User Login.....	80
II)	Signup.....	80
III)	Forgot Password	80
IV)	Identify the human face and recognize that person	80
V)	Sending notifications on the user's android devices	80
VI)	Displaying notifications data to the user	81
VII)	Data is saved into the database	81
26)	Approach.....	82
1)	Unit testing:	82
a.	Branch Testing:.....	82
b.	Graphical User Interface (GUI)Testing:	82
c.	Gorilla Testing:	82

2)	Integration Testing:.....	82
a.	Component Testing:.....	82
b.	Integration Testing:.....	82
c.	Incremental Integration Testing:	82
3)	System Testing:	83
a.	Functional Testing:.....	83
b.	Recoverability Testing:.....	83
c.	Performance Testing:.....	83
4)	Acceptance Testing:	83
a.	User Acceptance Testing:.....	83
b.	Operational Acceptance Testing:.....	83
c.	End-to-End Testing:.....	83
27)	Suspension and Resumption	84
a)	Suspension Criteria.....	84
b)	Resumption Requirements.....	84
28)	Testing Materials (Hardware / Software Requirements).....	85
a)	Hardware Requirements	85
b)	Software Requirements.....	85
29)	Test Cases.....	86
27.1)	Android Application Test Cases	86
27.1.1)	Android Application Login.....	86
27.1.2)	Android Application Forgot Password	88
27.1.3)	Android Application Create Account	90
27.1.4)	Android Notification When the Application is in Background	93
27.1.5)	Android Notification When the Application is in foreground.....	95
27.2)	Web Application Test Cases	97
27.2.1)	Upload Data on the Server.....	97
27.2.2)	Modify Data:	99
27.2.3)	Delete Data:	100
27.2.4)	Recognition:	101
27.2.5)	Web Application Login:.....	102
27.2.6)	Employee Forgot Password:	104
30)	Testing Schedule	106
V.	Project Issues	107
31)	Open Issues	107

32)	Off-the-Shelf Solutions	108
30.1)	Ready-Made Products	108
30.2)	Reusable Components.....	108
30.3)	Products That Can Be Copied.....	108
33)	New Problems	109
31.1)	Effects on the Current Environment	109
31.2)	Effects on the Installed Systems	109
31.3)	Potential User Problems.....	109
31.4)	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product.....	110
31.5)	Follow-Up Problems	110
34)	Tasks.....	111
32.1)	Project Planning:.....	111
1.	Requirement Gathering:	112
2.	Analysis:	112
3.	Design:.....	112
4.	Coding:	112
5.	Testing:.....	113
6.	Deployment:.....	113
32.2)	Planning of the Development Phases.....	113
1.	Planning:	113
2.	Analysis:	114
3.	Design:.....	114
4.	Coding:	114
5.	Testing:.....	114
6.	Deployment:.....	114
35)	Migration to the New Product	115
33.1)	Requirements for Migration to the New Product.....	115
33.2)	Data That Has to Be Modified or Translated for the New System.....	115
36)	Risks	116
37)	Costs	116
38)	Waiting Room	117
39)	Ideas for Solutions.....	117
40)	Project Retrospective.....	118
41)	Conclusion.....	118

42)	Future Enhancement.....	118
43)	References.....	119

Table of Figures

3.2	Working of Proposed System	16
8.1	System Environment	23
9.3.1	Employee Use Case	33
9.3.2	Admin Use Case	35
10.3.1	Data Objects	37
10.3.2	Complete Data Model	37
21	System Architecture	48
21.1	Overall Design	49
21.2	Class Diagram	50
21.3.1	Activity Diagram	51
21.3.2	Sequence Diagram	52
21.4	Subsystem Decomposition	53
21.5	Hardware Software Mapping	54
21.7	Persistent Data Management	56
22	Screenshots.....	66
32.1	Software Development Life Cycle	111

Table of Tables

3.3	Work Partitioning	17
6	Naming Conventions and Definitions	21
8.3.1.1	Employee	25
8.3.1.2	System Admin	25
8.3.1.3	Worker/Workers	26
8.3.2.1	Admin Log in Use Case	26
8.3.2.2	Register of Employee Use Case	27
8.3.2.3	Deletion of Employee Use Case	27
8.3.2.4	Modification of Employee's Data Use Case	28
8.3.2.5	Employee Log-In Use Case	28
8.3.2.6	Employee Password Generator Use Case	29
8.3.2.7	Employee Upload Data Use Case	29
8.3.2.8	Employee Delete/Modify Data Use Case	30
21.6.1	Fraud Person Data	55
21.6.2	Employee	55
21.6.3	Admin	55
27.1.1	Android Login	86
27.1.2	Android Forgot Password	88
27.1.3	Android Create Account	90
27.1.4	Android Notification When the Application is in Background	93
27.1.5	Android Notification When the Application is in foreground	95
27.2.1	Upload Data on the Server	97
27.2.2	Modify Data	99
27.2.3	Delete Data	100
27.2.4	Deletion and Recognition	101
27.2.5	Django Login	102
27.2.6	Django Forgot Password	104
28	Testing Schedule	106

I. Project Description

1) Project Overview

The technology aims in imparting a tremendous knowledge oriented technical innovation these days. Deep Learning is one among the interesting domain that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. Pre-trained models are used as the starting point to save on compute and time resources. Nowadays stealing, damaging the goods in a shopping mall becomes common. The fraud persons steal the goods and left the shopping malls undetected, after they were gone, the responsible authority comes to know about the stealing or damaging of the goods. Sometimes the damage is done either knowingly or unknowingly. This results in huge losses to the persons who are running these businesses.

In general, the surveillance system in order to catch these fraud persons can be classified into two types broadly:

- Manual Surveillance System
- Automated Surveillance System

Manual Surveillance System is a process where you keep a guard to catch the fraud person, if he/she revisits to your mall (or) shop after sometime. The problem arises over here is memory, how can human being can be able to remember the faces of these goons, if they come after a long time. Another problem is, the guard might not pay full attention to the incoming people at the entrance and the fraud person comes in the mall (or) shop undetected. To solve all these issues, we go with Automatic Surveillance System

Automatic Surveillance System is a process to automatically estimate the presence or the absence of the fraud person/persons at the entrance by using facial recognition technology. Our software is based on the above process and it also provides a notification to the employees who are running our android application in their smartphones.

2) The Purpose of the Project

2.1) The User Business or Background of the Project Effort

Nowadays stealing, damaging the goods in a shopping mall becomes common. The fraud persons steal the goods and left the shopping malls undetected, after they were gone, the responsible authority comes to know about the stealing or damaging of the goods. Sometimes the damage is done either knowingly or unknowingly. This results in huge losses to the persons who are running these businesses. The intention behind our software is to identify those fraud person or thief if he/she revisits that place. That fraud person might had done some damage to the goods of the mall or some act of stealing and left the mall without getting caught.

2.2) Goals of the Project

The main goal of our software is to identify the fraud person when they revisit the places like shopping malls, departmental stores and many more such kind of places. It will identify and recognizes only those people who had done some damage to the properties or steal something from those places in the past. After identifying and recognizing the faces of those people, it will notify the workers about that person by sending the image of that person and some description about that person on the workers android devices.

2.3) Measurement

The main objectives of our software are to reduce the training time that required to train a person's image. And our software requires a single image of a person to train a model and it identifies and recognizes a person with more than 90% accuracy.

3) The Scope of the Work

3.1) The Current Situation

The traditional facial recognition system requires more than 100 images of a person to train a model. The more images we give to the model the more accuracy we get when recognizing the face of a person. Due to this, the training time required to train a person's model increases and it also requires a lot of computation power. As the software requires more images to train and to store the images, there is a high probability that the system may run out of space problem. To get up to 80-85% accuracy, we need to give a minimum of 100 images to the software. And moreover, when the software identifies and recognizes any person, it will notify only to the workers who are monitoring to that place. It is not possible for the current system to notify about the identified and recognized person to the workers.

3.2) The Context of the Work

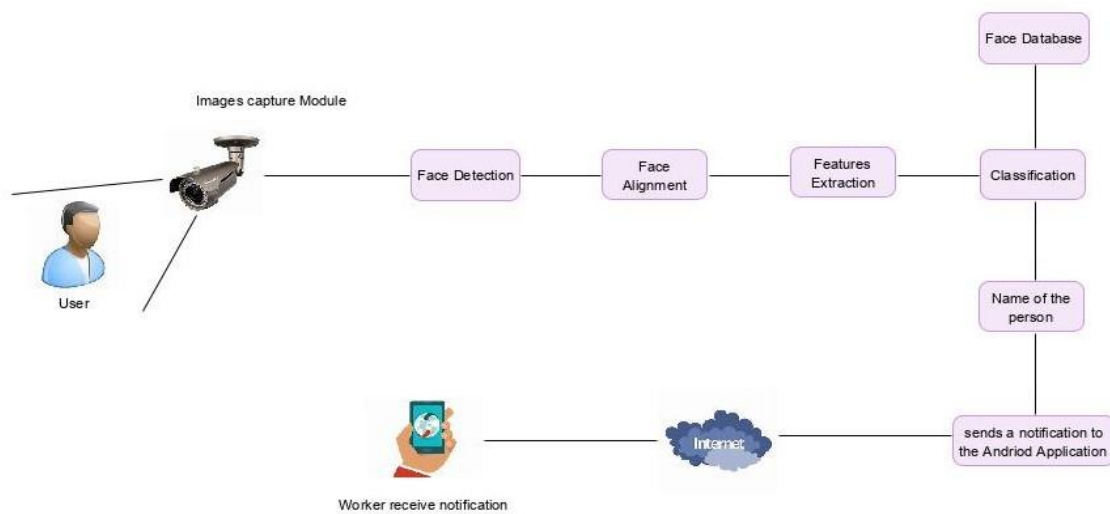


Figure-3.2: Working of Proposed System

The above figure shows working of the proposed system of “**Facial Recognition System**” and shows how the work flows. Whenever, any person visits the places like shopping malls, departmental stores and any other places where our software can be used, the software monitors the person through a surveillance camera and check for existing record. If record is found, it sends a notification to the workers who are working in that place. In notification, it will send some details regarding that person such as the image of a person and how much damage that person has done in past and what things that person had damaged and many more such details are mentioned in the notification.

3.3) Work Partitioning

By understanding the current system problems and has decided some solutions that can help us to overcome with this problem, we have divided our project into two parts:

1. Django Web Application (for admin)
2. Android Application (for admin and for employees)

The table below shows the modules that we have implemented in our software.

Event	Input	Output	Summary
1. Login	1. User Name 2. Password	Login success (or) failure.	Allow login to only those people who has valid user name and password.
2. Signup	1. First Name 2. Last Name 3. Date-of-Birth 4. Gender 5. Phone Number 6. Email 7. Password 8. Confirm Password	Account created successfully (or) account creation failed.	Allow new users to create as account.
3. Train a Person's Model	1. Name 2. Image	Train a person's model.	Allow admin to train a person's model.

	3. Loss 4. Description		
4. Send Notification	1. Name 2. Image 3. Loss 4. Description	Send notification to the workers/employees.	When our software identifies and recognizes the face of any fraud person, it notifies to the admin and allows admin to send notifications to the workers/employees.
5. Android Login	1. User Name 2. Password	Login success (or) failure.	Allow login to only those people who has valid user name and password.
6. Android Signup	1. First Name 2. Last Name 3. Date-of-Birth 4. Gender 5. Phone Number 6. Email 7. Password 8. Confirm Password	Account created successfully (or) account creation failed.	Allow new users to create as account.
7. Android Receive Notification When the Application is in Background or killed	Receives the notification data that Django application has send	Display notification data to the user.	Allow android users to read the data what Django application (or) admin has send.
8. Android Receive Notification When the Application is in Foreground	Receives the notification data that Django application has send	Display notification data to the user.	Allow android users to read the data what Django application (or) admin has send.

3.4) Competing Products

The traditional facial recognition system requires more than 100 images of a person to train a model. The more images we give to the model the more accuracy we get when recognizing the face of a person. Due to this, the training time required to train a person's model increases and it also requires a lot of computation power. As the software requires more images to train and to store the images, there is a high probability that the system may run out of space problem. To get up to 80-85% accuracy, we need to give a minimum of 100 images to the software. And moreover, when the software identifies and recognizes any person, it will notify only to the admin who is monitoring to that place. It is not possible for the current system to notify about the identified and recognized person to the workers. This software's can be used in the places like **shopping malls, departmental stores**, and many other similar places.

4) Stakeholders

4.1) The Client

The shopping malls owners are the clients for this project. They might want to invest money on the production of our software.

4.2) The Customer

The shopping malls owners are the customers to our software. They might want to use our software to catch the fraud persons and increase their profit.

4.3) Hands-On Users of the Product

The users shall know English language, because our software comes with the default language that is English. Our software doesn't require trained persons, in order to use the software.

4.4) Priorities Assigned to Users

- **Key users:** System Admin which can be the manager of the shopping mall is the key user of the software
- **Secondary users:** The employees are considered as the secondary users of our software.
- **Unimportant users:** The workers are considered as the unimportant users as their work is to receive the notification and take appropriate action accordingly.

4.5) Maintenance Users and Service Technicians

These are the trained people usually programmers who are responsible for the maintenance of software.

5) Mandated Constraints

5.1) Solution Constraints

The product shall recognize the person, if only proper picture of the picture fed to the software.

5.2) Implementation Environment of the Current System

The product is developed on windows 10 platform and might work on windows 8 or later. The product shall run in the environment where the TensorFlow version is 1.10.0. The reason for the specific version of the TensorFlow is use of opensource model which is developed using this version of TensorFlow.

5.3) Partner or Collaborative Applications

There are such no collaborative application or applications which our product is dependent on.

5.4) Off-the-Shelf Software

The product shall use the open source deep learning model, which is optimized to recognize the person face after training the model with the single image of the same person. The product shall use the open source library called MTCNN for detection and face alignment.

5.5) Schedule Constraints

The time allowed to make this product is 3 months approx.

5.6) Budget Constraints

There are no such constraints were applied while building of the software.

6) Naming Conventions and Definitions

Term	Definition
System Admin	A person who is responsible for adding, updating or deleting of worker, employee
Employee	A person who is responsible for adding, deleting or modifying of worker's details and he is also responsible for enrolling an image of the fraud person
Worker	A person who is using android application, he will receive a notification about the presence of the fraud person in the mall
Fraud Person	A person who has committed a crime in the shopping mall or stole the goods or damaged the goods of the mall in the past
Face Database	Collection of all the information about the image of a fraud person.
Stakeholders	Any person who has contributed to the project
Face embedding	The deep learning model analyses an image and returns numerical vectors that represent each detected face in the image in a 128-dimensional space
MTCNN	Multi-Task Cascaded Convolutional Neural Networks is a neural network which detects faces and facial landmarks on images

7) Relevant Facts and Assumptions

i) **Methodology:**

- Project will follow waterfall methodology throughout execution.
- Project will follow team governance guidelines and requirements.

ii) **Technology:**

- The team will write the solution in python.
- The software will be a web application which will be made using Django framework.
- The product will send the notification to the android application via Restful services.

iii) **Schedule:**

- Materials will arrive as planned within the project schedule.

iv) **Installation:**

- The system needs to installed the appropriate library which are mentioned in the product material.

II. Project Description

8) Relevant Facts and Assumptions

8.1) Use Case Diagrams

This application, Fraud Detection at Shopping Malls using Facial Recognition System is shown in the below figure. Firstly, a person's image should be provided using the camera. Then, camera transfers it to the Face Recognition System. Next, an image is pre-processed for features of face. After that, they are compared with an image in the system database and if matched, the system sends a notification to the worker.

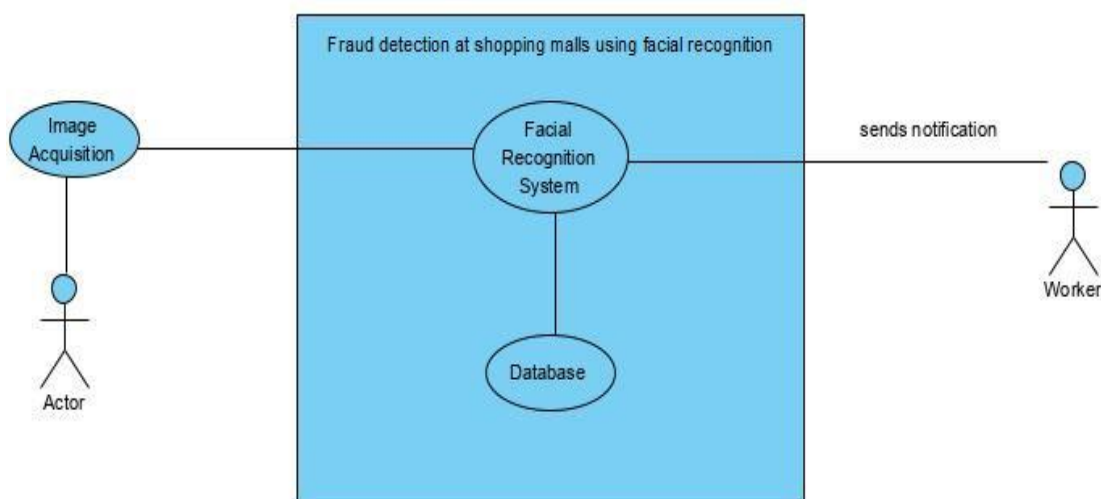


Figure-8.1: System Environment

8.2) Product Use Case List

Specification of Actors (Employee)	25
Specification of Actors (Admin)	25
Specification of Actors (Worker)	26
Use case 1 – Admin login	26
Use case 2 - Register of Employee	27
Use case 3 - Deletion of Employee	27
Use case 4 - Modifying of Employee's Data	28
Use case 5 - Employee Log-In	28
Use case 6 - Employee Password Generator	29
Use case 7 - Employee Upload Data	29
Use case 8 - Employee Delete/Modify Data	30

8.3) Individual Product Use Cases

8.3.1) Specification of actors

In the analysis stage of the Fraud Detection at Shopping Malls using Facial Recognition System development process, actors below are described so far.

8.3.1.1) Employee:

Employee	
Element	Details
Description	An Employee, who is responsible for feeding or uploading the data in the software.
Examples	An Employee, who captures the face of the person and upload it to the Servers.

8.3.1.2) System Admin:

System Admin	
Element	System Admin
Description	A system admin is a person who handles the application.
Examples	System admin who login in the system and manages the operations regarding the Employees like adding a new Employee, deleting the Employees and Updating the details of the Employee.

8.3.1.3) Worker/Workers:

Worker	
Element	Worker
Description	A worker is a person who receives the notification.
Examples	When the person's face matches with the face which are present in the database, the system sends a notification which is receive by the worker via android application that helps him to keep an eye on the person.

8.3.2) Specifications of Use Cases

8.3.2.1) Admin Log in Use Case:

Login	
Element	Details
Actor	System Admin
Trigger	System Admin wish to start using the system
Pre-Conditions	System Admin must not log into the system
Post Conditions	System Admin is logged into the system and the system menu is displayed
Normal Course	1) System Admin fills the username and password 2) System Admin click the login button

8.3.2.2) Register of Employee Use Case:

Register	
Element	Details
Actor	System Admin
Trigger	System Admin after click on the Add Employee Tab with all the details of the Employee
Pre-Conditions	The Employee is not registered
Post Conditions	The Employee is registered and cannot be re-registered
Normal Course	1) System Admin fills the data in the form 2) System Admin stores the data in the database

8.3.2.3) Deletion of Employee Use Case:

Deletion	
Element	Details
Actor	System Admin
Trigger	When there is need to delete the Employee
Pre-Conditions	The Employee should be registered
Post Conditions	The Employee's data will be deleted from the database
Normal Course	1) System Admin select the Employee to be deleted 2) System Admin clicks the delete button

8.3.2.4) Modification of Employee's Data Use Case:

Modification	
Element	Modification
Actor	System Admin
Trigger	When there is need to modify the Employee's data
Pre-Conditions	The Employee should be registered
Post Conditions	The Employee's data will be modified in the database
Normal Course	1) System Admin clicks the modify button 2) Change the appropriate data 3) Click on Save Button

8.3.2.5) Employee Log-In Use Case:

Login	
Element	Details
Actor	Employee
Trigger	Employee wish to start using the system
Pre-Conditions	Employee must not log into the system
Post Conditions	Employee is logged into the system and the system menu is displayed
Normal Course	1) Employee fills the username and password 2) Employee click the login button

8.3.2.6) Employee Password Generator Use Case:

Password Generator	
Element	Details
Actor	Employee
Trigger	Employee wish to generate the password for his account
Pre-Conditions	Employee must not log into the system
Post Conditions	Password is generated and system login page is displayed
Normal Course	1) Employee fills the password 2) Employee click the Generate button

8.3.2.7) Employee Upload Data Use Case:

Upload Data	
Element	Details
Actor	Employee
Trigger	Employee wants to upload data to the server
Pre-Conditions	Employee must be logged into the system
Post Conditions	Data will be uploaded to the server and system menu is displayed
Normal Course	1) Employee fills the form with appropriate details 2) Employee click the Upload button

8.3.2.8) Employee Delete/Modify Data Use Case:

Delete/Modify Data	
Element	Details
Actor	Employee
Trigger	When there is need to delete/modify the existing data
Pre-Conditions	1) Employee must be logged into the system 2) Data must exist in the database
Post Conditions	Database will be updated after respective action performed
Normal Course	1) Employee clicks on delete/modify button 2) If delete button is clicked, the data will be deleted 3) Else Employee will fill the form and click on Save Button

9) Functional Requirements

9.1) Summary of Functions

The Fraud Detection at Shopping Malls using Facial Recognition requires a technology-based solution for Shopping malls whose primary functions include:

1. Authentication
2. User Accounts
3. Password Generator
4. Data Accessibility
5. Entity Profile Administration
6. Data Integrity
7. System Security

9.2) Functional Requirements

In order to accomplish the above articulated need, the FDP Fraud Detection at Shopping Malls using Facial Recognition requires a technology-based solution for Shopping malls that includes the following base functionality: (See below section 23 for system mock up).

1) Authentication

Authenticating the user credentials with the database, so that authorized person can use the system with respective designation.

Example: An Employee cannot be login as system admin in the system.

2) User Accounts

1. User accounts are password protected.
2. Password resets are handled by the website.

3) Password Generator

Password Generator will allow the user to generate the password of his own, when the user login for the first time.

4) Data Accessibility

1. Information is available for the authorized person respectively.
2. An Employee can access the data from the database about the fraud persons.
3. A system admin can access the employee's data.
4. The respective user shall be logged in before data accessibility.

5) Entity Profile Administration

1. Authorized users can update the entity's profile directly on the website.
2. The respective user shall be logged in before performing the update/delete operation.

6) Data Integrity

1. System sends a notification when the face matches to the android application.
2. If feasible, system will also provide alert/notification upon sign-in by entity user when any of the above dates or data elements requires updating.

7) System Security

1. User accounts are password protected.
2. Data can be access by the authorized person.
3. Modification to the data can be done by the authorized person.
4. Before performing any sort of operation, the user should be logged in.
5. The user cannot access any webpage with the URL certain conditions are applied to the respective user.

9.3) Use Cases

9.3.1) Employee Use Case

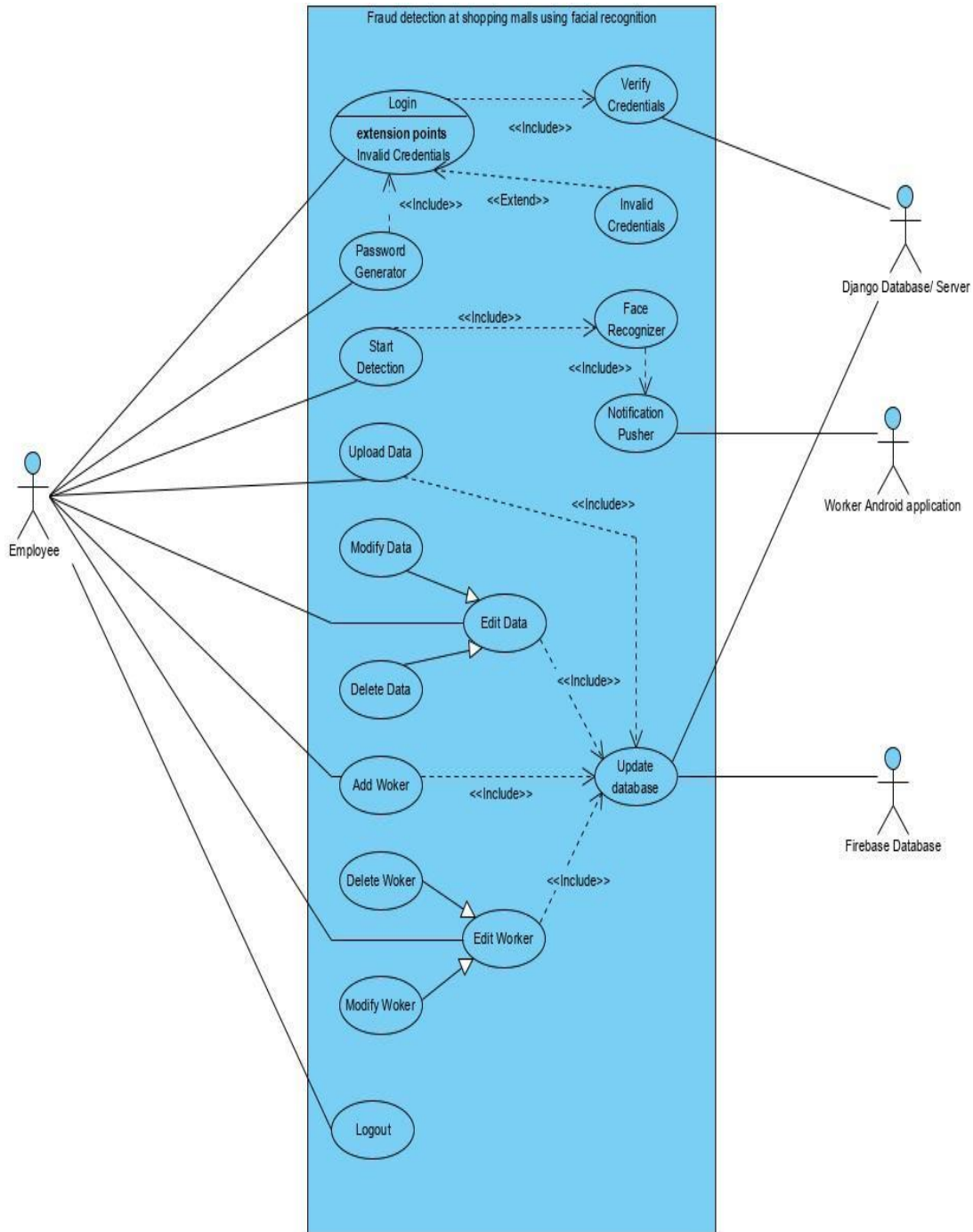


Figure-9.3.1: Employee Use Case

- Employee can generate password, if he is logging in for the first time
- Employee can login in the system
- Employee can add data by the filling the form
- Employee can update/delete the data
- Employee can add a Worker
- Employee can update/delete the Worker's profile
- Employee can Log Out from the System

9.3.2) Admin Use Case

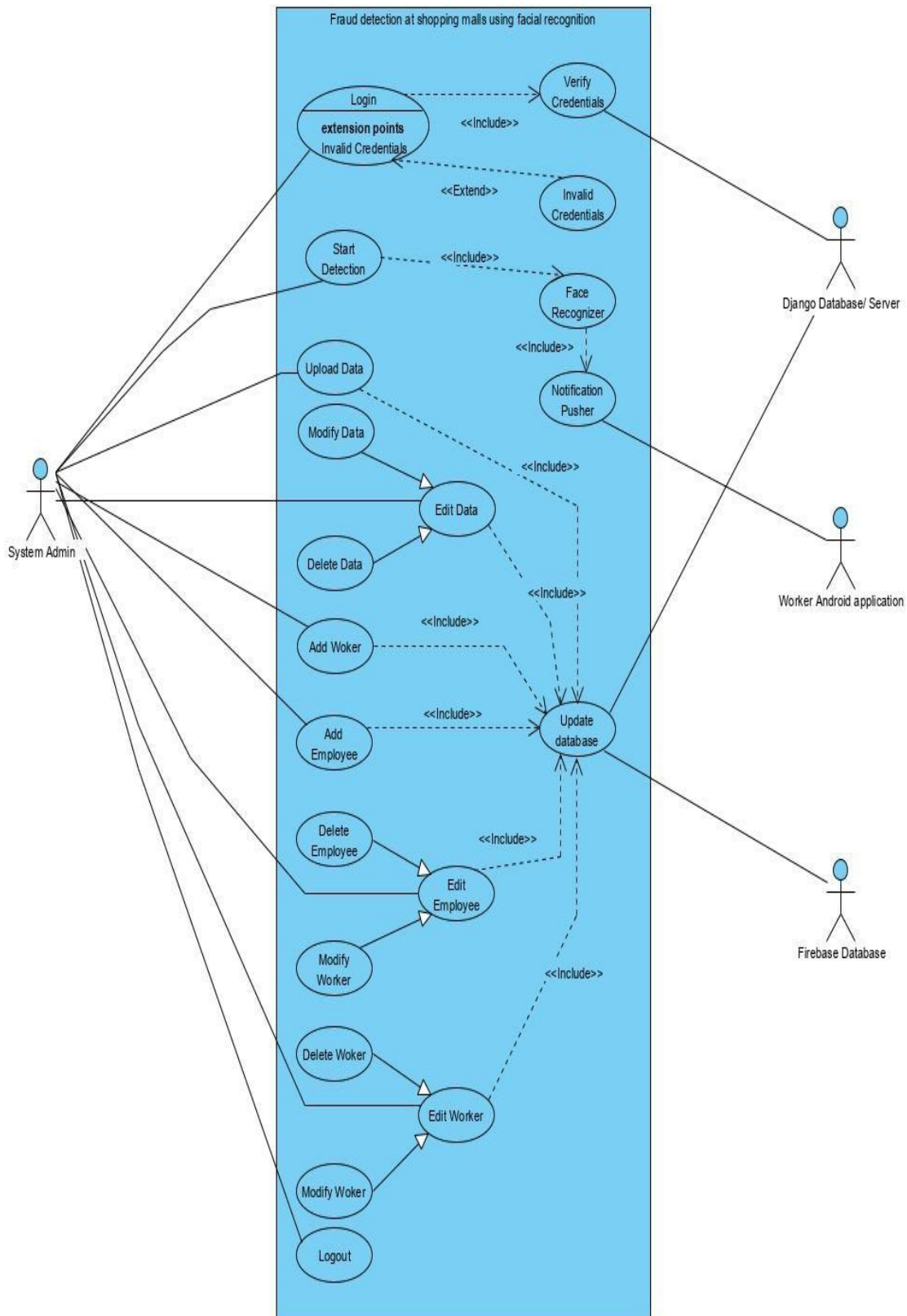


Figure-9.3.2: Admin Use Case

- Admin can login in the system
- Admin can add data by the filling the form
- Admin can update/delete the data
- Admin can add a Worker
- Admin can update/delete the Worker's profile
- Admin can add Employee
- Admin can update/delete the Employee's profile
- Admin can Log Out from the System

10) Data Requirements

10.1) Data Description

Basically, 3 types of data objects will be manipulated by the software. These data objects are namely System Admin object, Employee object and Worker object.

10.2) Data Objects:

- a) System Admin object: This object contains information about the admin, the information it holds are username and password.
- b) Employee object: This object holds information about the employee which includes email id, password, date of birth etc.
- c) Worker Object: This object holds information about the worker which includes username, password with some other details.

10.3) Relationships



Figure-10.3.1: Data Objects

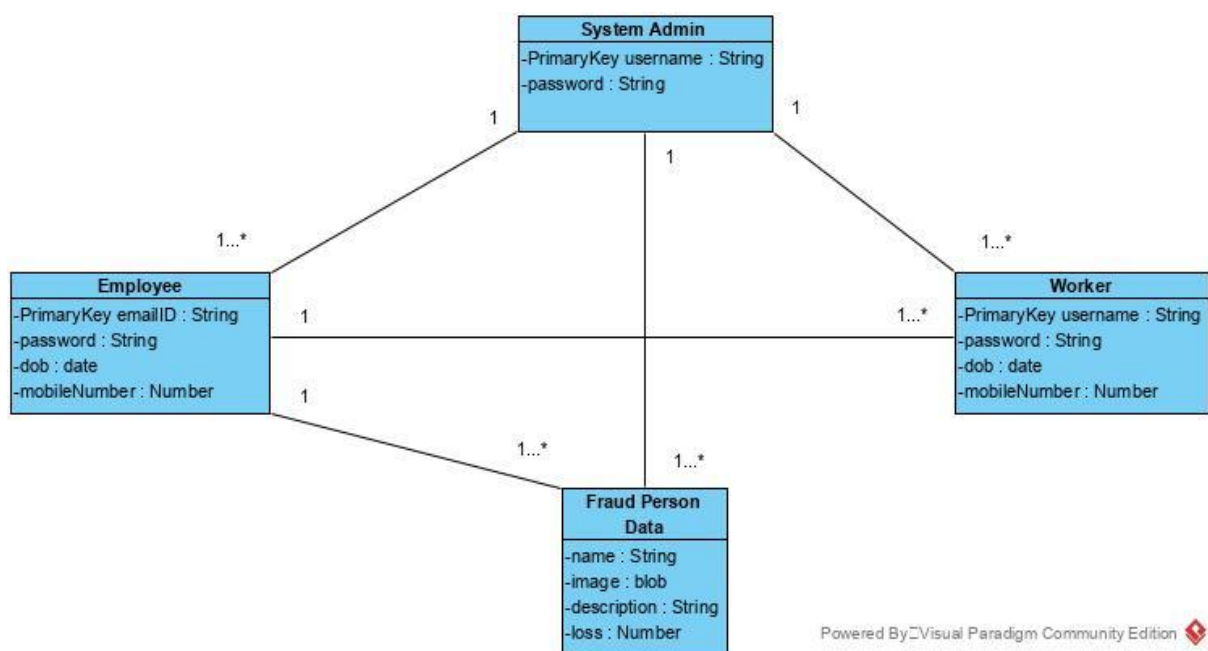


Figure-10.3.2: Complete Data Model

The admin object has control over all the objects. A single admin object can handle more than employee object, worker object as shown below. An employee object can handle more than one object of fraud Person Object. A worker object has only access to the details of the fraud person object.

11) Performance Requirements

11.1) Speed and Latency Requirements

Camera's visual must run smoothly without any error and delay more than 4sec to get the image of the user. This requirement is depended on many aspects of the user personal computer. Minimum requirements for running Fraud detection at shopping malls using facial recognition are:

- GPU: Intel HD Graphics
- CPU: Intel Pentium
- Camera: Minimum 2MP Camera
- USB port: 1x USB 2.0 or better port

Operating system: Windows XP or better.

11.2) Precision or Accuracy Requirements

The system will employ numerous data quality assurance techniques, including but not limited to:

- Inputs masks
- Drop down lists with standard responses
- Record data completeness requirements
- Basic data logic warnings (e.g., submitting without fillings all fields, email validation, password strength checker, etc.).

11.3) Capacity Requirements

The system is intended to be available online 24 hours per day, 365 days per year with the exception of scheduled and pre-notified system maintenance downtimes, if needed.

The system also gets halted when it is requires to generate face embeddings for the new data. Although the time required is relatively small probably more or less 30 seconds per face. Again, the time depends on some factors which include pc specifications.

Data will become immediately available for use; except for the new data which has to be undergone the process to give the face embedding which then will be appended to the existing data.

12) Dependability Requirements

12.1) Reliability Requirements

- **Stability:** the product didn't cause crashes, unhandled exceptions or script errors.
- **Data Integrity:** all types of data remain intact throughout the product. As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.
- **Trustworthiness:** the product's behaviour is consistent, predictable and trustworthy.

12.2) Availability Requirements

- System will be available while the computer is running on. But when there is need to process the new information then it will halt for a more or less 30 seconds per person data.
- The internet services shall not get disrupted while sending the information to the server. Although irrespective of the internet, the system will be available for information will not be send to the server again in case of loss.

12.3) Robustness of Fault-Tolerance Requirements

- The system cannot stand by, if the camera is not working properly. For now, our system only supports single camera supervision and in case of damage or fault to it will halt the system
- System stores the data not only in the SQLite database but in the firebase as well, which makes it more robust. In case of any failure to any one of database, we can grab data from another database.

- In terms of power sources, the organization can have power generator to provide the power to the system and the camera which is doing surveillance. In this way the system become more robust to power failures.

12.4) Safety-Critical Requirements

- If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to server like firebase and reconstructs a more current state by reapplying or redoing the operations.

13) Maintainability and Supportability Requirements

13.1) Maintenance Requirements

- As a tool to obtain the ease of maintainability UML will be used in the development process.
- **Flexibility:** The system has the ability to change as required by the customer. As the whole system lightly coupled it becomes easier to change it in the future.
- **Extensibility:** It is possible to add features to the system if the customer asks.
- **Simplicity:** The code is not more complex than needed, and does not obscure test design, execution and evaluation.
- **Readability:** The code is adequately documented and easy to read and understand.
- **Modularity:** The code is split into manageable pieces.
- **Analysability:** It is easier to find causes for defects in the system, although various exceptional conditions are already handled in the system.

13.2) Supportability Requirements

- **Identifiers:** It is easy for the customer to understand different parts of the system, it doesn't require training in order to start using the system.
- **Versatility:** The system has the ability to be use in more than one way than it was originally designed.

- **Troubleshoot able:** The system doesn't design to be trouble shoot by itself. It is better to approach the programmer or any knowledgeable person.
- **Debugging:** The system is not design to be able to debug by the customer himself.

13.3) Adaptability Requirements

- The product is expected to run under Windows 10 with some specific versions of TensorFlow 1.10.1 and python 3.6.9
- The product is designed to run in shopping malls, but with some changes in the system. It can even run in schools as an attendance machine, it can also be use a surveillance system by the government.

13.4) Scalability or Extensibility Requirements

- This product is capable of processing the 10,000 existing entries from the database. This number might be getting increase to 50,000 may be in a year.
- This product shall process 50,000 entries from the database.

13.5) Longevity Requirements

- The product shall be expected to operate within the maximum maintenance budget for a minimum of three years.

14) Security Requirements

14.1) Access Requirements

- Only authorized users can be able access their accounts since accounts access is password protected.
- System Admin can access all the employees' details and can modify them accordingly.
- Employee cannot access the other employee account.
- Employee access all the data related fraud person.
- System Admin does not have the access to the Employee's password.

14.2) Integrity Requirements

- The system prevents itself from being produced incorrect data.
- It does not allow anyone to access webpage via the URL, thus protect itself from being use abuse intentionally.

14.3) Privacy Requirements

- The product ensures the privacy of the Employees and Workers.
- Example: The system admin cannot see the password which is generated by either Employee or Worker.
- Employee cannot see the Workers' password, although he is responsible for adding the Workers' details to the system.

14.4) Immunity Requirements

- Since the product is made in Django, it has some built-in security features like, Cross site scripting (XSS) protection, Cross site request forgery (CSRF) protection, SQL injection protection, Click jacking protection, SSL/HTTPS, Host header validation.
- The above-mentioned security features will allow the product to with stand against some malicious attacks.

15) Usability and Humanity Requirements

15.1) Ease of Use Requirements

- It will be easy for the user to become familiar with and competent in using the user interface during the first contact with the website. For example, if the user wants to access the data from the database, the user should be able to move through the sequence of actions to display data.
- It will be easy for users to achieve their objective through using the website. For example, if the user has the goal to add data to the database then there are very simple steps that can be followed to perform the action.
- It will be easy to recall the user interface and how to use it on subsequent visits. The steps involved to perform various actions are very simple and can be recall easily.

15.2) Personalization and Internationalization Requirements

- The product comes with a default language English, which most widely spoken language.
- The expected loss that a fraud person causes is registered in rupees which are Indian currency.

15.3) Learning Requirements

- The product shall be used by a member which is at least completed Xth class.
- The product shall be used by a member who at least knows English language.

15.4) Understandability and Politeness Requirements

- Well, there is not any specific understandability requirements since there are no symbols are used in the product.
- The product shall hide the functionality of the processes from the user.

15.5) Accessibility Requirements

- The user should not have any disability related to sight, hands or legs.

15.6) User Documentation Requirements

- Documentation will be provided for the persons who are responsible for the maintenance.
- Although the end user will not require any documentation to use the application.

15.7) Training Requirements

- There are no training requirements to use the product.

16) Look and Feel Requirements

16.1) Appearance Requirements

- The product shall comply with corporate branding standards.
- The look and feel of the product are friendly, approachable, professional and stylish.

16.2) Style Requirements

- The transitions when mouse hover the anchor tags shall be not fast enough and should be less than 2 seconds long.
- The font size of the all tags shall be of standard size.
- The image size shall be in range 150X150 pixels.

17)Operational and Environmental Requirements

17.1) Expected Physical Environment

- The product shall be used by a worker, sitting down on a chair, inside of an office.

17.2) Requirements for Interfacing with Adjacent Systems

- The simulation requires a camera and computer hardware. Also, it requires HDMI and USB port.

17.3) Producing Requirements

- The product shall be distributed in ZIP file.
- The product shall be installed by a professional person probably a programmer after reading installation guide.
- Before installing the product, the required libraries must be installed with the specific version by the professional.

17.4) Release Requirements

- Each release shall not cause previous features to fail.
- Each release might want to upgrade specific library to specific version.

18) Cultural and Political Requirements

18.1) Cultural Requirements

- The product shall not be offensive to religious or ethnic groups.
- The product comes with the default language which is English.

18.2) Political Requirements

- There are not such requirements for the use of this product.

19) Legal Requirements

19.1) Compliance Requirements

- Personal information shall be implemented so as to comply with the Data Protection Act.

III. Design

20) System Design

20.1) Design Goals

20.1.1) From the Perspective of Users

- **Usability:** It should be easy to use this system. It comes with all necessary functionality.
- **User-friendly Interface:** The system should come with simple user interface
- **Low Cost:** We intend to provide this software as service to different shopping malls. The additional cost will be internet cost.
- **Security:** Our software is more secure, as it does show the password of employee or worker to the admin which makes it more secure.

20.2) From the Perspective of us

- **Performance:** The application uses web service database; hence the requests and responses should be shortened which leads to better performance.
- **Memory:** It must provide efficient queries to the database in order to use consume less RAM.
- **Updatability:** Personal records, fraud person data will be updated frequently.

21) Proposed Software Architecture

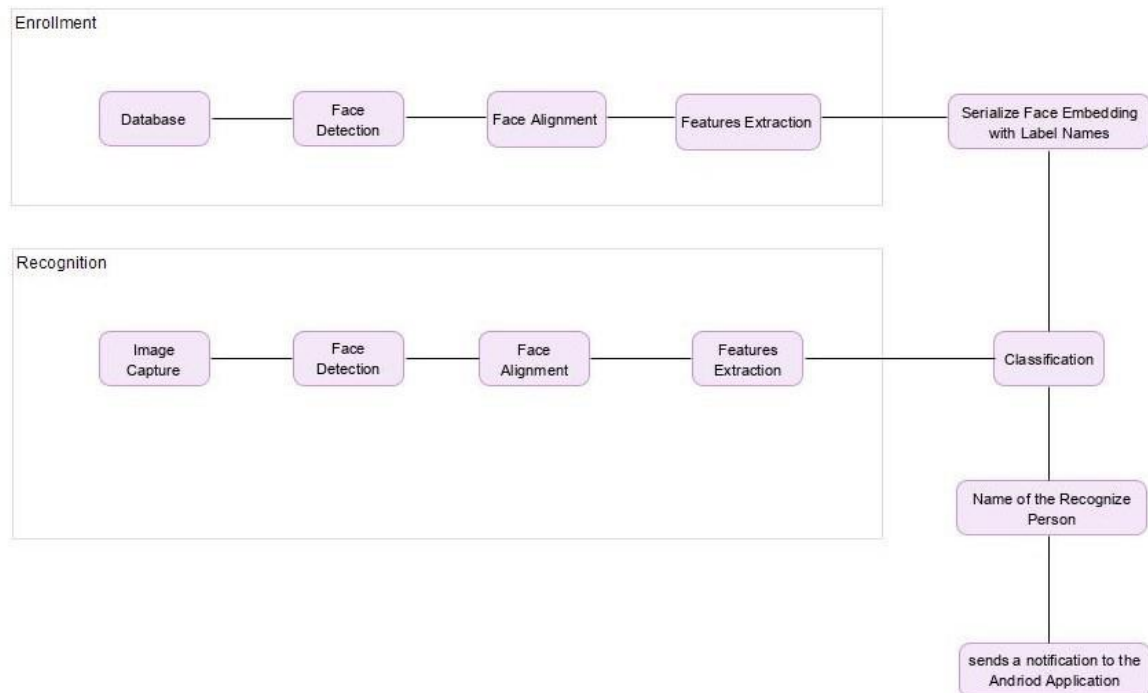


Figure-21: System Architecture

Our system uses facial recognition system to aware the workers about the presence of the fraud persons in their mall by means of a notification. The system is made with the intention to be use in shopping malls. It combines the software, android application and camera. The system composes of face acquisition module, face recognition and notification module mainly.

The design process of the system can be divided into two stages: training or enrolment and recognition. Among them, the main task of the training stage is to process the collected face images and provide face embeddings for the face database. The identification stage is much the same as the former process, but the purpose is different. The specific steps are as follows:

First, the image acquisition device acquires the face image of the visitor; second, the collected face image is pre-processed and the feature extraction process is carried out to generate the face embedding data of the visitor. Thirdly, the face embedding feature data of visitors are matched one by one with the data items of face database. If the matching is successful, then the system sends a notification to all the worker, who are using our android application.

21.1) Overview

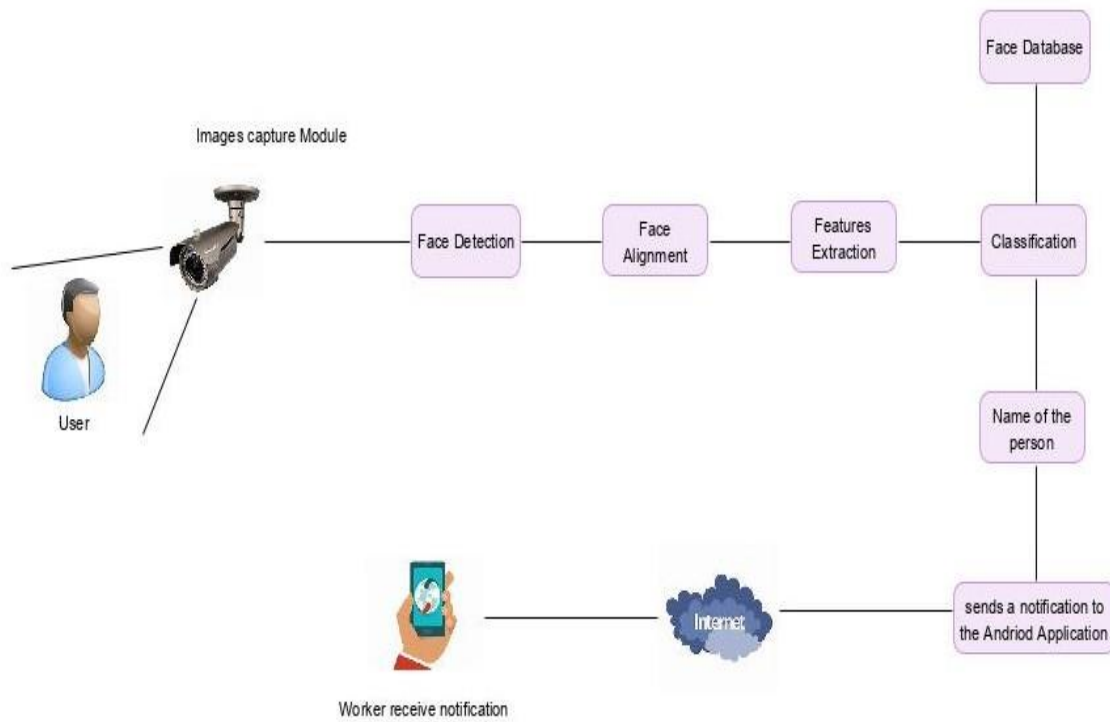


Figure-21.1: Overall Design

The system designed composed of hardware and software. Among them, the hardware includes camera, remote android phone, software includes recognition algorithm and database.

The main work of the system is to collect face images. After a series of processing, the face images are matched with the face data in the database, and the notification will be sent to the users who are using android application with the correct matching results. The face image acquisition of the system is completed by the camera.

21.2) Class Diagram

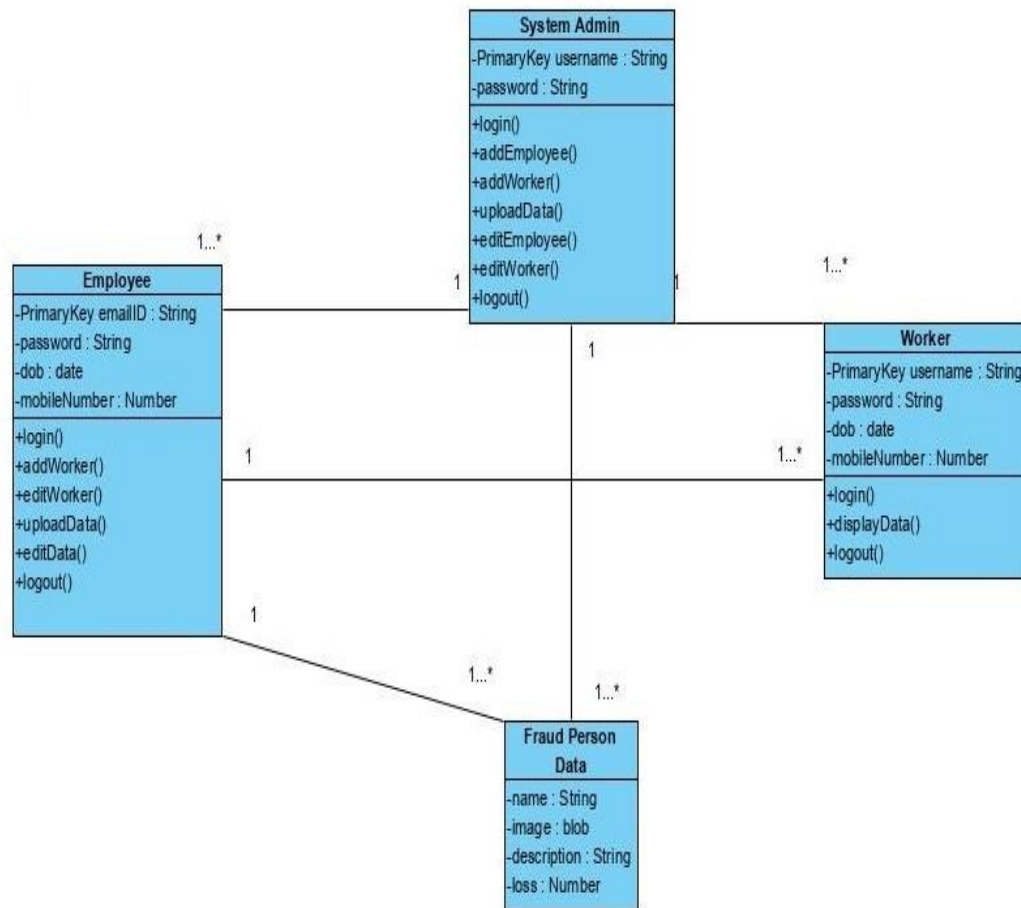


Figure-21.2: Class Diagram

As shown in the above figure, the main classes of the system are shown above. The admin class is the core class of the system. It can have one to many relationships with Employee class, Worker class and face database. The Employee class have the one to many relationships with the face database. The worker class also have the one to many relationships with the face database.

21.3) Dynamic Model

21.3.1) Activity Diagram

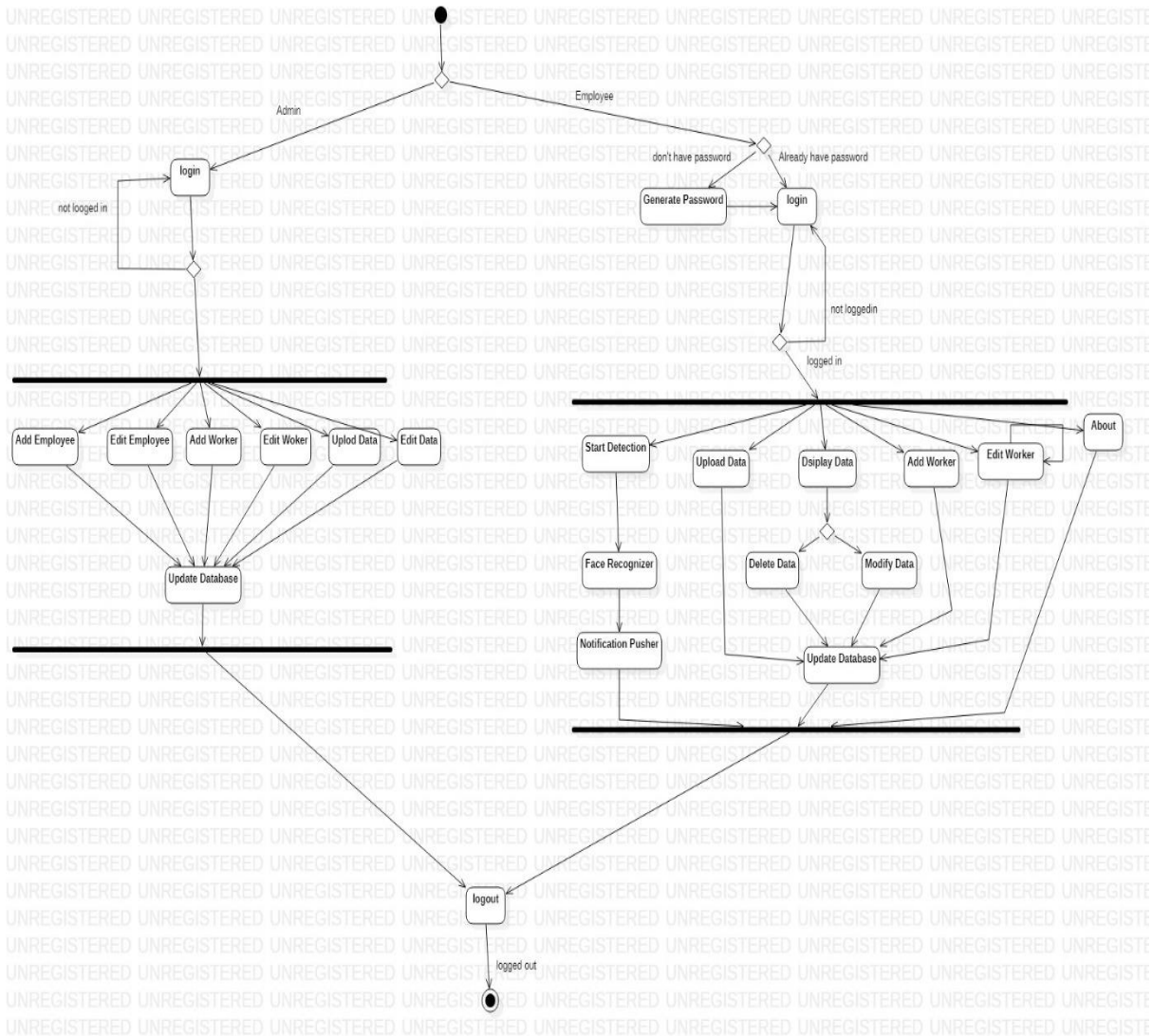


Figure-21.3.1: Activity Diagram

The system allows the employee and system admin to login. There are various operations that admin can perform like add Employee, edit Employee etc. The system allows the employee to generate password if he has doesn't have one. After the generation of password, the employee can login in the system to perform various operations like adding Worker, edit Worker, upload data regarding the fraud person, display data etc. Lastly, he can logout of the system.

21.3.2) Sequence Diagram

The below diagram shows the sequence of steps that system admin encounters while using the system.

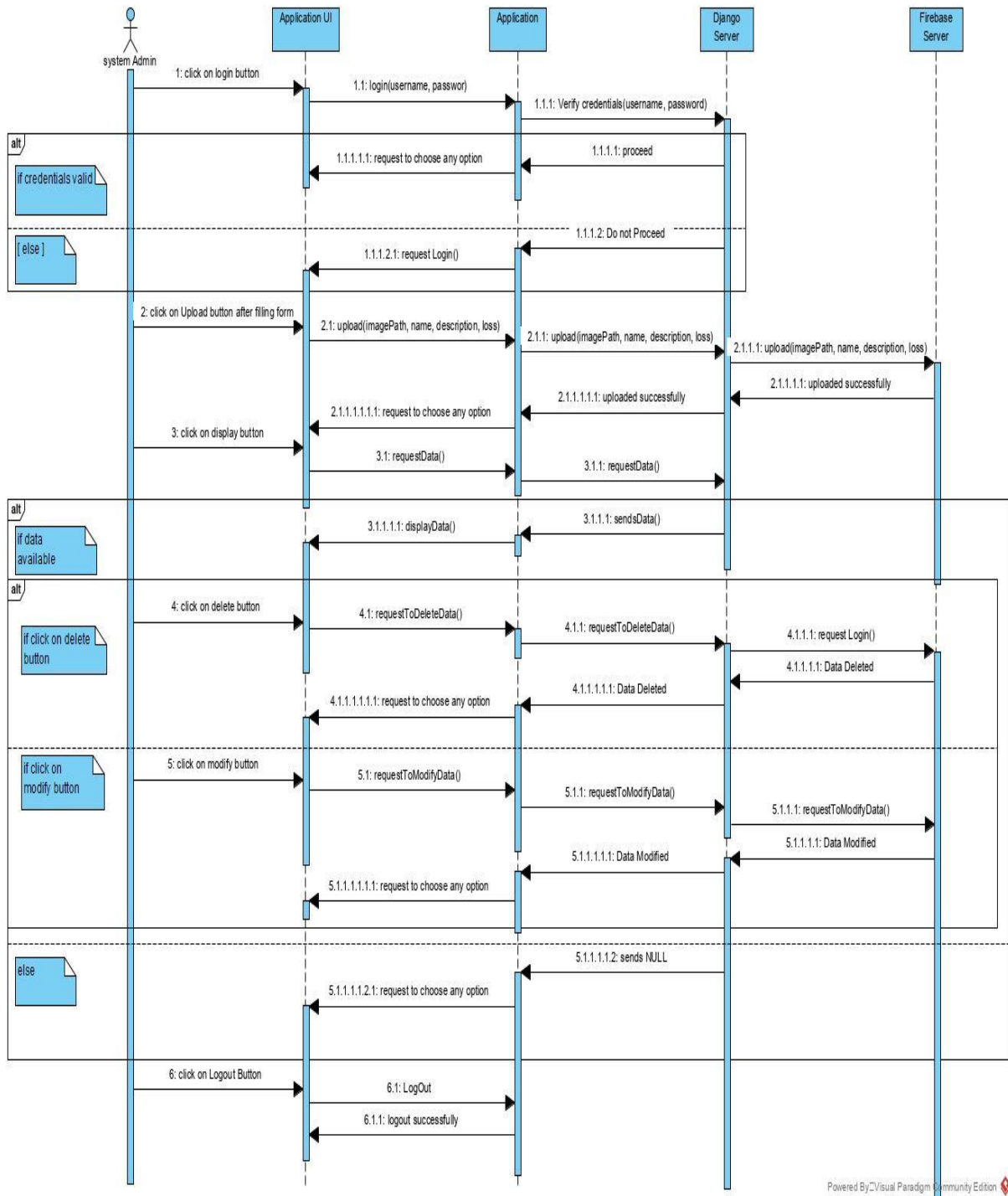


Figure-21.3.2: Sequence Diagram

21.4) Subsystem Decomposition

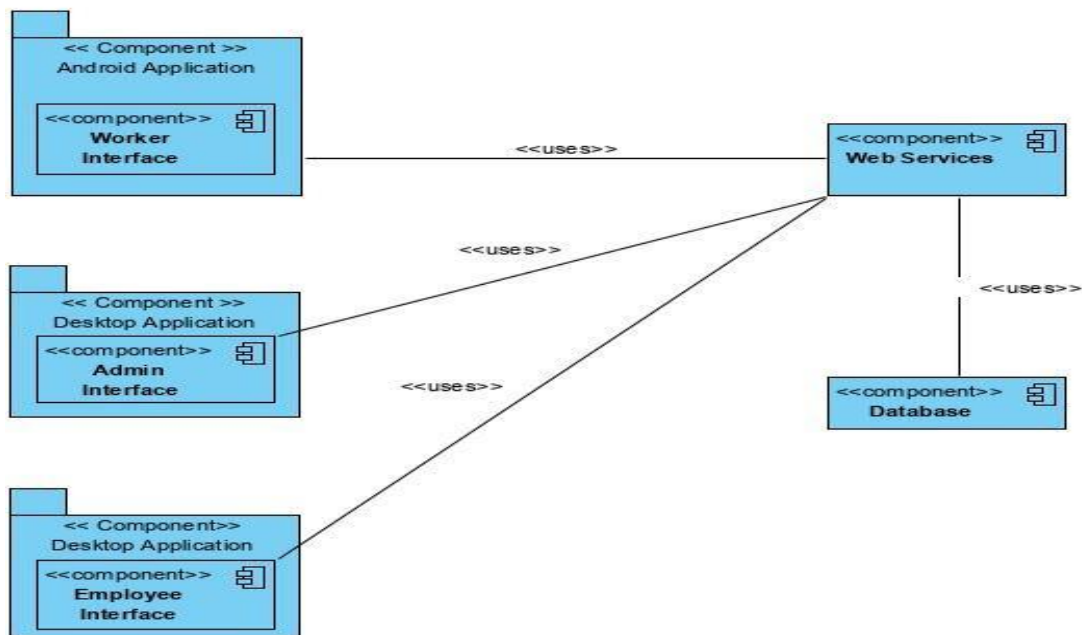


Figure-21.4: Subsystem Decomposition

The desktop application is divided into two subsystems, the Admin interface and Employee interface. These components are implemented using Django. The Worker interface is an android application and it is implemented using java.

The web service provides the connection between the application server and database. We use Django as web service in our project. In addition, it provides all the core functionalities of our application like registration of Employee, registration of worker, uploading data, editing data etc.

Workers will connect to the application by using an android device while the admins and employees can connect to the application using their PC. Both the worker, the admins and the employees will use the web services to connect to the database and insert/delete information. For the web services we use Django MVT, RESTful Services, JavaScript and jQuery. The database server is SQLite.

21.5) Hardware / Software Mapping

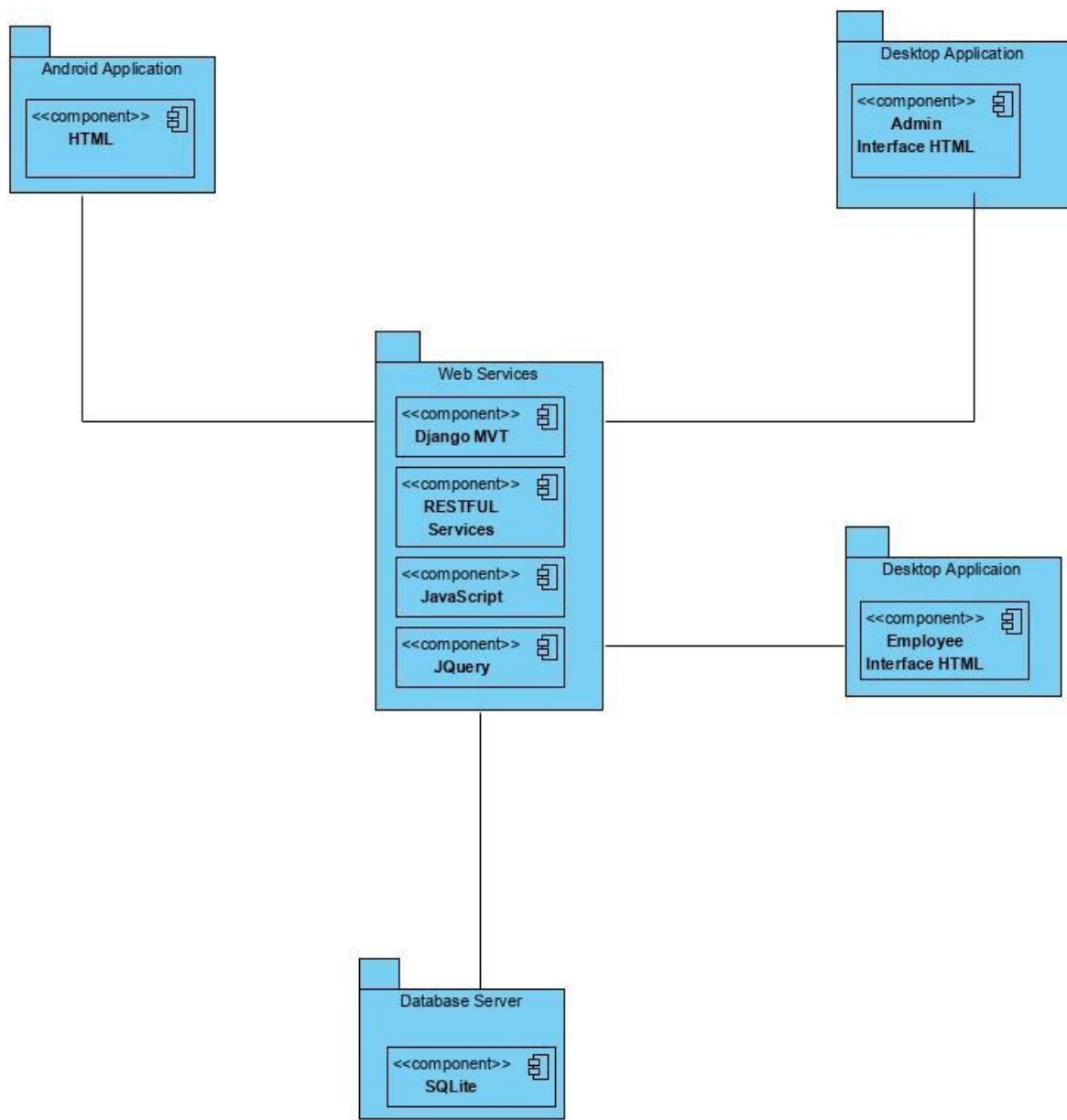


Figure-21.5: Hardware Software Mapping

Worker will connect to the application by using an android device while the admins and employee can connect to the application using their PC. Both the worker, employee and the admins will use the web services to connect to the database and insert/delete information. For the web services we use Django MVT, RESTful Services, JavaScript, and jQuery. The database is SQLite.

21.6) Data Dictionary

21.6.1) Fraud Person Data

Field	Type	Max Value	Null	Key	Default
customID	IntegerField		No	Primary Key	
name	charField	100	No		Name
Image	ImageField		No		
description	TextField	200	Yes		
loss	IntegerField		no		1000

21.6.2) Employee

Field	Type	Max Value	Null	Key	Default
empid	IntegerField		No		
email	CharField	100	No	Primary Key	
password	PasswordField	50	Yes		
mobileNumber	IntegerField		No		
data	CharField	100	no		

21.6.3) Admin

Field	Type	Max Value	Null	Key	Default
username	CharField	100	No	Primary Key	
password	CharField	50	No		

21.7) Persistent Data Management

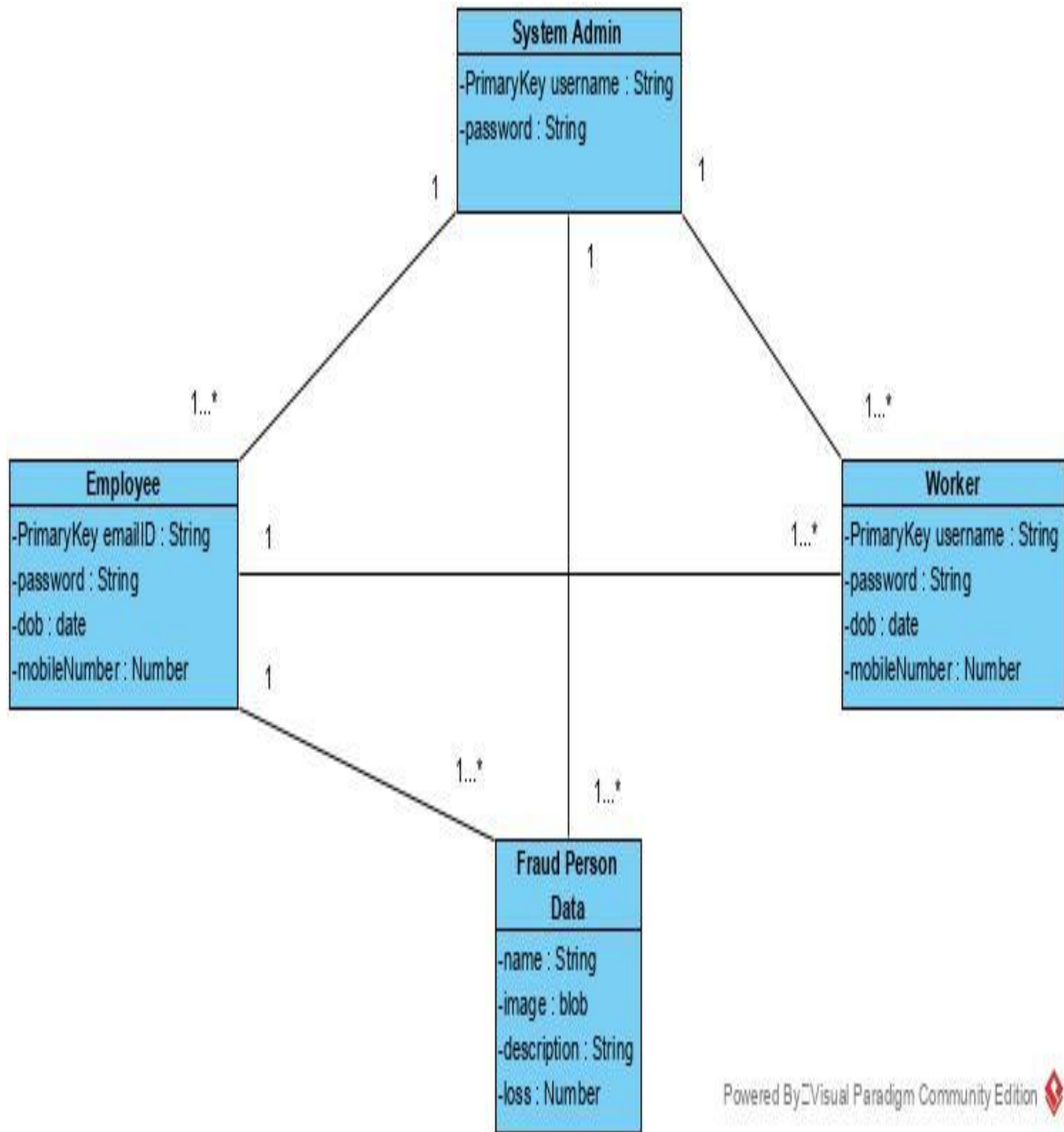


Figure-21.7: Persistent Data Management

The above figures show a physical persistent model which describes a relational database's schema.

21.8) Access Control and Security

The access control is provided by first defining users and then assigning them various access rights. The user management is done through a Java class which allows adding users, defining login information, changing the login information, and deleting the users.

The user management can only be performed by the person who created the database. A user can be allowed/refused following five kinds of rights:

- **Access:** The user is enabled/disabled to read persistent objects.
- **Update:** The user is enabled/disabled to write persistent objects.
- **Delete:** The users are enabled/disabled to delete persistent objects.
- **All:** The user is enabled/disabled to read, write, and delete persistent objects.
- **Grant:** The user is enabled/disabled to perform all the previous operations and to give grant permission.

22) Implementation

Step-by-step process for any facial recognition process: The major steps in any facial recognition process are to:

- find the face of the person in the image;
- analyse the facial features of the person;
- compare the analysed features of the persons with the known faces; and
- predict the name of the person based on the above comparisons.

Finding the face/faces of the person/persons in an image:

```
def extract_face(filename, detector, required_size=(160, 160)):
    print(detector)
    from PIL import Image
    face_array = list()
    if ".ipynb_checkpoints" in filename:
        print(filename)
        pass
    else:
        image = Image.open(filename)
        image = image.convert('RGB')
        pixels = np.asarray(image)
        detector1 = MTCNN()
        results = detector1.detect_faces(pixels)
        x1, y1, width, height = results[0]['box']
        x1, y1 = abs(x1), abs(y1)
        x2, y2 = x1 + width, y1 + height
        face = pixels[y1:y2, x1:x2]
        image = Image.fromarray(face)
        image = image.resize(required_size)
        face_array = np.asarray(image)
    return face_array
```

The filename is the path of the image and the size of the image shall be 160X160, otherwise the function compresses the size of the image to be 160X160.

Analysis the facial features of the crop image:

```
def image_to_embedding(image, model):
    image = cv2.resize(image, (160, 160))
    face_pixels = image.astype('float32')
    print(face_pixels)
    mean, std = image.mean(), image.std()
    image = (image - mean) / std
    samples = expand_dims(image, axis=0)
    face_embedding = model.predict(samples)
    return face_embedding[0]
```

This function will convert the raw image into face embeddings, which is an array of 128 values which represents the face from the image.

Comparing the analysed features of the persons with the known faces and Predict the name of the person based on the above comparisons:

```
def recognize_face(face_image, input_embeddings, loaded_model):
    embedding = image_to_embedding(face_image, loaded_model)
    minimum_distance = 200
    name = None
    id = -1
    for (label_name, list_of_images) in input_embeddings.items():
        for image in list_of_images:
            euclidean_distance = np.linalg.norm(embedding - image)
            print('Euclidean distance from %s is %s' % (label_name, euclidean_distance))
            if euclidean_distance < minimum_distance:
                minimum_distance = euclidean_distance
                actualName = label_name.split("_")[0]
                id = label_name.split("_")[1]
                # print("ID: {}".format(id))
                name = actualName
                # print(name)
    if minimum_distance < 10:
        if str(actualName) not in i:
            print("pushing notification for : {}".format(actualName))
            imageObject = Image.objects.get(customId=id)
            imageName = imageObject.name
            description = imageObject.description
            loss = imageObject.loss
            pushNotification(id, imageName, description, loss)
            i.append(actualName)
        return str(name + " " + str(round(100 - minimum_distance, 3)))
    else:
        return "Unknown"
```

The input embedding in the function, are the face embedding of the known faces which are going to be compare with the dynamic generated face embedding and if the similarity of the two faces embedding is greater than the threshold value then a notification will be send to the android application.

Push Notification for Android Application:

```
def pushNotification(id, name, description, loss):
    from pusher_push_notifications import PushNotifications
    beams_client = PushNotifications(
        instance_id='5d6090ee-2b02-4f62-9dfc-0e475f3b54ba',
        secret_key='6F8EA217E680B1915207B2E4B8095E782AC4C538F4777A2B0C858C3A976C9840'
    )
    response = beams_client.publish_to_interests(
        interests=['hello'],
        publish_body={
            'apps': {
                'aps': {
                    'alert': 'Something!'
                }
            },
            'fcm': {
                'notification': {
                    'title': 'Notification Arrived',
                    'body': 'ID: ' + str(id) + ' Name:' + name + ' Description: ' + description + ' Loss:' + str(loss)
                }
            }
        }
    )
    print(response)
    print(response['publishId'])
```

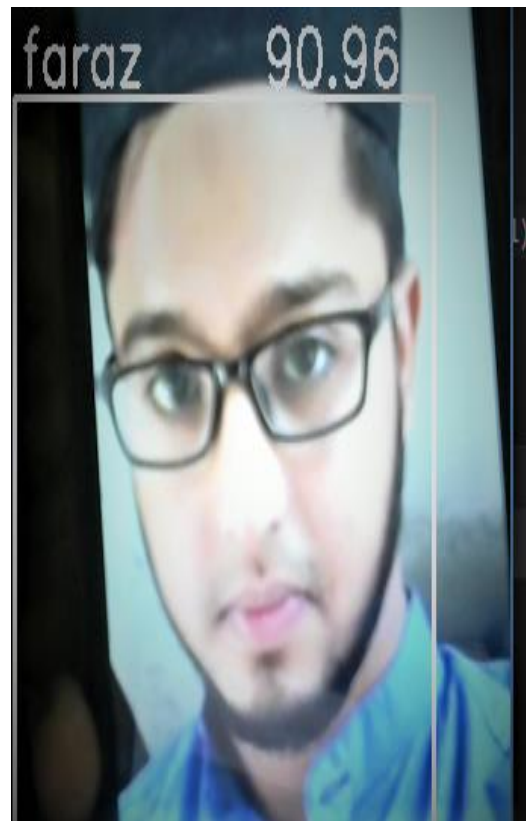
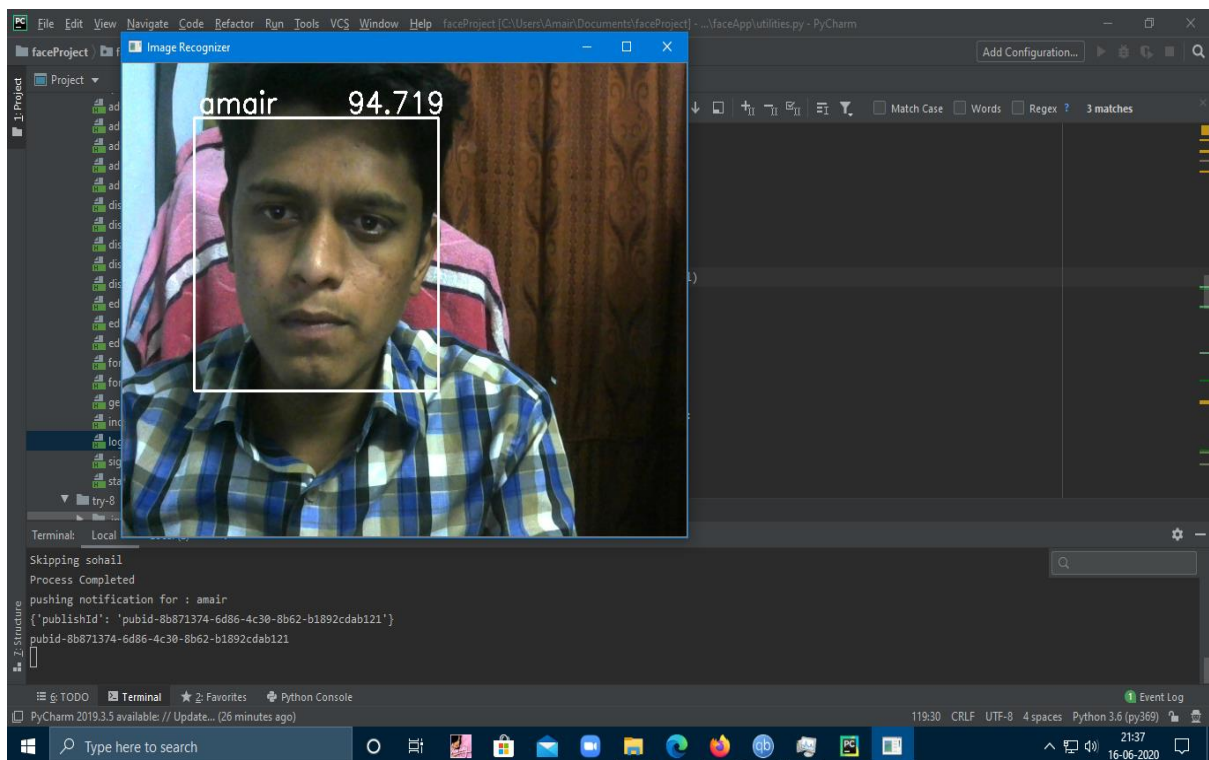
The above code is use to send notification to the android device, when the face is recognized in the live video feed.

The overall system is implemented in Python 3.6.9

The images of fraud persons in the database are use as training dataset. The images from the database one by one used to extract the face embeddings (code shown above). Then the face embeddings will be serialized in the local file system.

At the time of Recognition, the face embeddings which were serialized, will be deserialize and will be compared to the face embeddings of the dynamically generated face embeddings from live video. If the two faces matches, then the system will push a notification to the android application otherwise it returns unknown person as name.

Result:



Implementation of Notification in Android Application

Creating Notification Channel to Handle the Notifications in Android:

```
1 package com.example.displaynotificationandroid;  
2  
3 import android.app.Application;  
4 import android.app.NotificationChannel;  
5 import android.app.NotificationManager;  
6  
7 public class App extends Application {  
8  
9     public static final String FCM_CHANNEL_ID = "FCM_CHANNEL_ID";  
10  
11     @Override  
12     public void onCreate() {  
13         super.onCreate();  
14  
15         if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O){  
16             NotificationChannel fcmChannel = new NotificationChannel(  
17                 FCM_CHANNEL_ID, name: "FCM_Channel", NotificationManager.IMPORTANCE_HIGH  
18             );  
19  
20             NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
21  
22             manager.createNotificationChannel(fcmChannel);  
23         }  
24     }  
25 }  
26
```

Android Notification Channel

This class is responsible for creating a notification channel. And with the help of this class, the user can of android application can send and receive the notifications.

Handling Notification Data and Customizing Notification:

```
38  @Override
39  public void onMessageReceived(@NonNull RemoteMessage remoteMessage) {
40      super.onMessageReceived(remoteMessage);
41
42      Log.d(TAG, "msg: " + "onMessageReceived: Message Received From " + remoteMessage.getFrom());
43
44      if (remoteMessage.getNotification() != null){
45          String title = remoteMessage.getNotification().getTitle();
46          String body = remoteMessage.getNotification().getBody();
47          String imageUrl = remoteMessage.getNotification().getImageUrl().toString();
48          Log.d(TAG, "msg: " + "Image Url: " + imageUrl);
49
50          PendingIntent contentIntent = PendingIntent.getActivity( context: this, requestCode: 0, new Intent( packageContext: this, MainActivity
51
52          Notification notification = new NotificationCompat.Builder( context: this, FCM_CHANNEL_ID
53              .setSmallIcon(R.drawable.ic_android_black_24dp)
54              .setContentTitle(title)
55              .setContentText(body)
56              .setColor(Color.BLUE)
57              .setTimeoutAfter(30)
58              .build();
59
60          NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
61          manager.notify( id: 1002, notification);
62
63          String demo_title = remoteMessage.getNotification().getTitle();
64          String demo_message = remoteMessage.getNotification().getBody();
65          String demo_imageUri = remoteMessage.getNotification().getImageUrl().toString();
66
```

Android Handling Notification Data and Customizing Notification

The class that is shown in above figure is used to handle the notification data. It handles data in two cases

- i) When the application is in background / killed.
- ii) When the application is in foreground.

In both the cases, it will handle data differently. Moreover, this class is also used to customized the notification.

Getting the Notification Data and Displaying the Notification Data:

```
69     @Nullable
70     @Override
71     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
72         View v = inflater.inflate(R.layout.fragment_notification, container, attachToRoot: false);
73
74         LocalBroadcastManager.getInstance(getContext()).registerReceiver(mHandler, new IntentFilter(action: "com.example.displaynotification"));
75
76         userImage = v.findViewById(R.id.userImage);
77         displayPersonName = v.findViewById(R.id.displayPersonName);
78         displayPersonLoss = v.findViewById(R.id.displayPersonLoss);
79         displayPersonDescription = v.findViewById(R.id.displayPersonDescription);
80
81         if (getArguments() != null) {
82             title = getArguments().getString(Arg_Title);
83             message = getArguments().getString(Arg_Message);
84             personName = getArguments().getString(Arg_Name);
85             loss = getArguments().getString(Arg_Loss);
86             description = getArguments().getString(Arg_Description);
87             image = getArguments().getString(Arg_Image);
88             info = getArguments().getInt(Arg_Number);
89         }
90
91         displayPersonName.setText(personName);
92         displayPersonLoss.setText(loss);
93         displayPersonDescription.setText(description);
94
95         if (image == null) {
96             Picasso.with(getContext()) Picasso
97                 .load(R.drawable.ic_android_white_24dp) RequestCreator
```

Android Getting Notification Data and Displaying It

The function in the above figure is responsible for getting the notification data from the `FCMMessagingClass` and to display the notification data as shown in figure-22(o).

Retrieving All the Notifications Data and Displaying in A List Form:

```
26 public RetrieveNotificationData(Activity context, List<StoreNotificationData> displayData){
27     = super(context, R.layout.listview_row, displayData);
28     this.context = context;
29     this.displayData = displayData;
30 }
31
32 @NonNull
33 @Override
34 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
35     LayoutInflater inflater = context.getLayoutInflater();
36
37     View listItems = inflater.inflate(R.layout.listview_row, root: null, attachToRoot: true);
38
39     ImageView imageView = listItems.findViewById(R.id.imageView1ID);
40     TextView text1 = listItems.findViewById(R.id.nameTextViewID);
41     TextView text2 = listItems.findViewById(R.id.infoTextViewID);
42
43     StoreNotificationData storeData = displayData.get(position);
44
45     text1.setText(storeData.getNotifyPersonName());
46     text2.setText(storeData.getNotifyLoss());
47     Picasso.with(getContext()) Picasso
48         .load(storeData.getNotifyImage()) RequestCreator
49         .placeholder(R.drawable.ic_android_black_24dp)
50         .into(imageView);
51
52     return listItems;
53 }
54 }
```

Android Retrieving Notification Data and Displaying It in A List Form

In the above figure, the first function RetrieveNiotificationData is responsible for retrieving the notification data that user has received if past and store it into an array. And the next function is responsible for displaying all the notifications that the user has received in the past in the form of a list as shown in Figure-22(n).

23) Screenshots

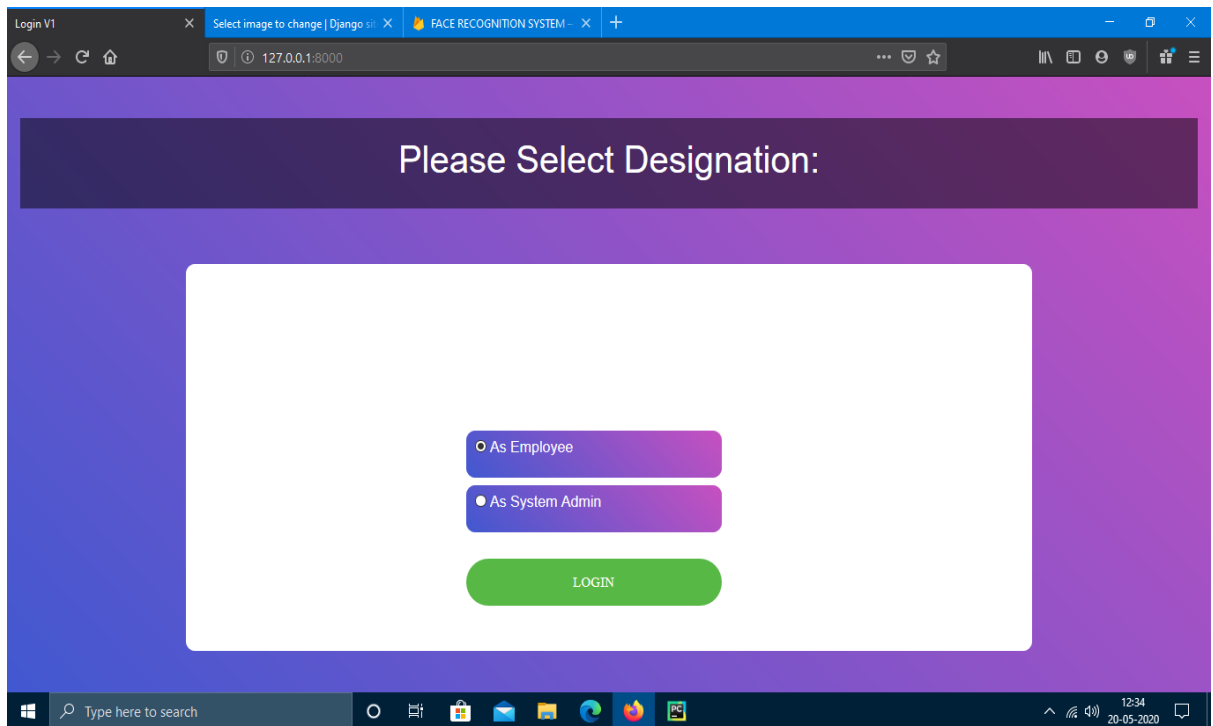


Figure 0 Login Screen to select Designation

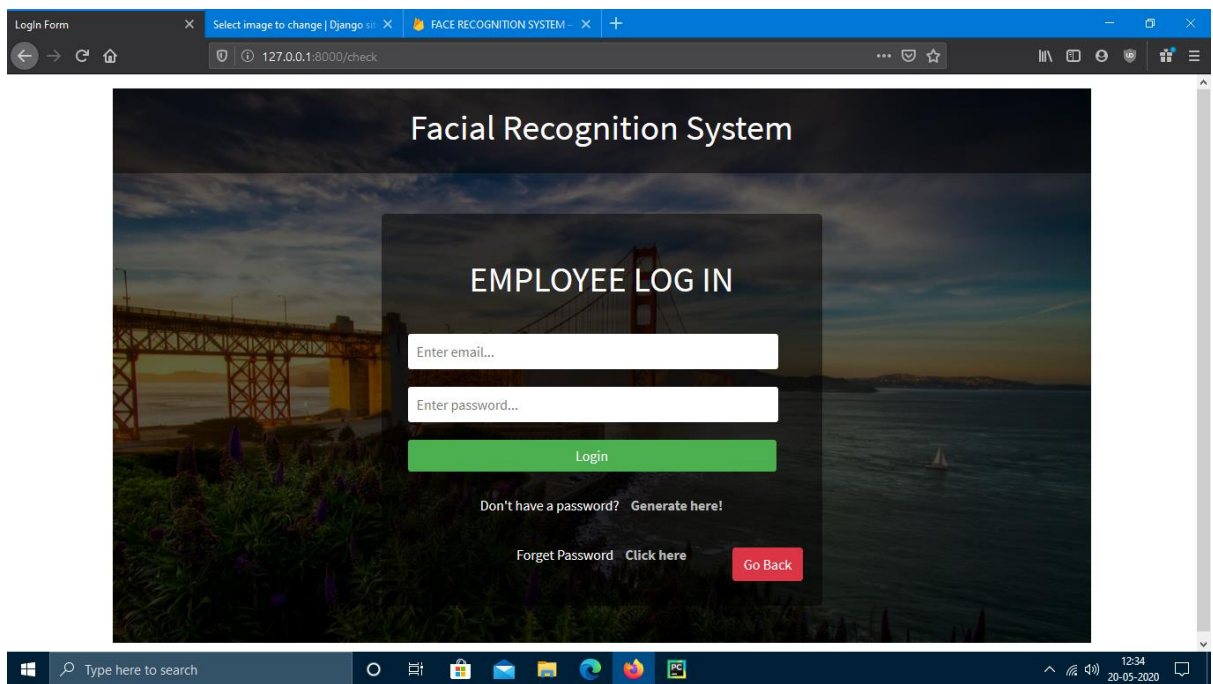


Figure 1 Login Screen for Employee to Login

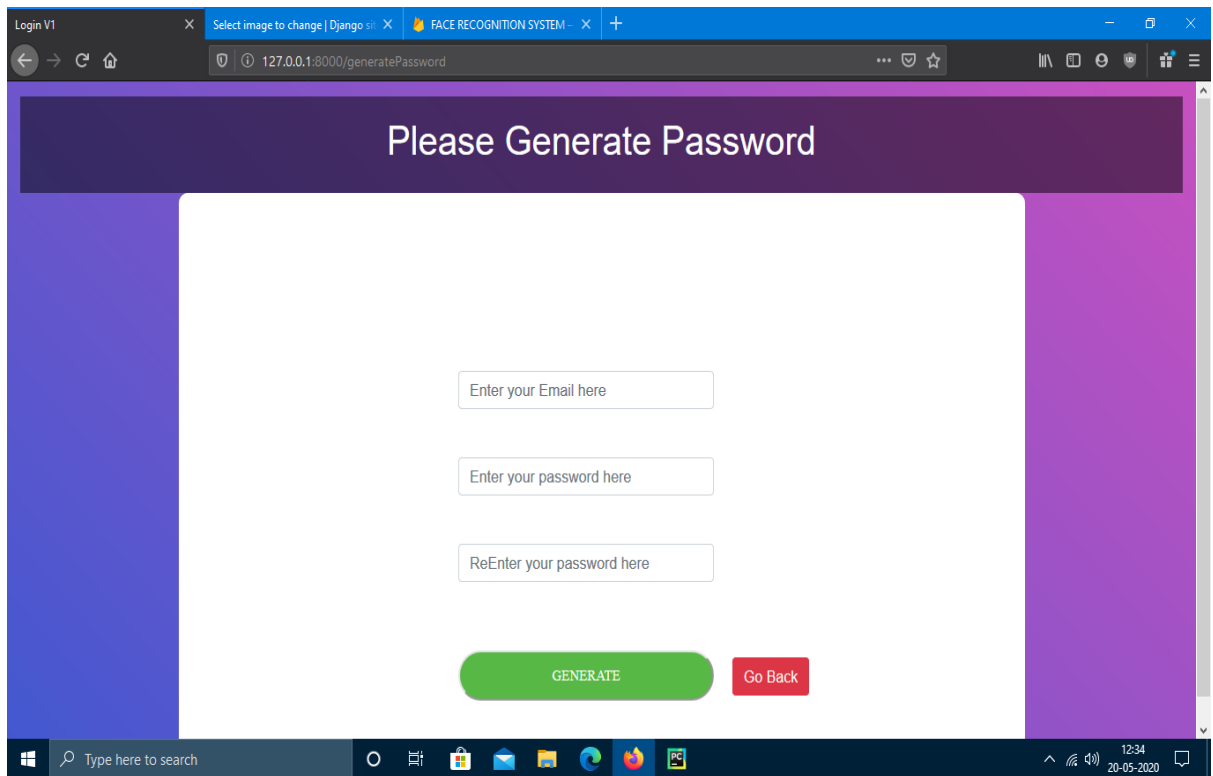


Figure 2 Password generator for Employee

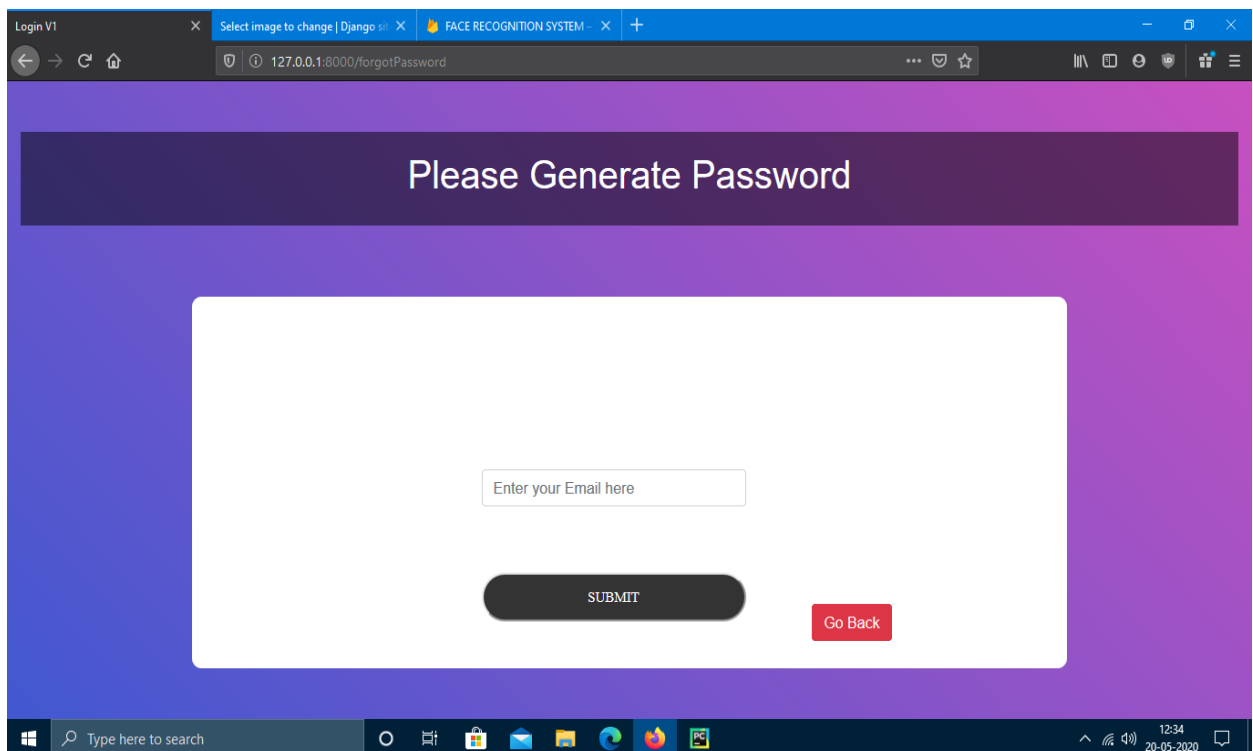


Figure 3 Forget Password for Employee

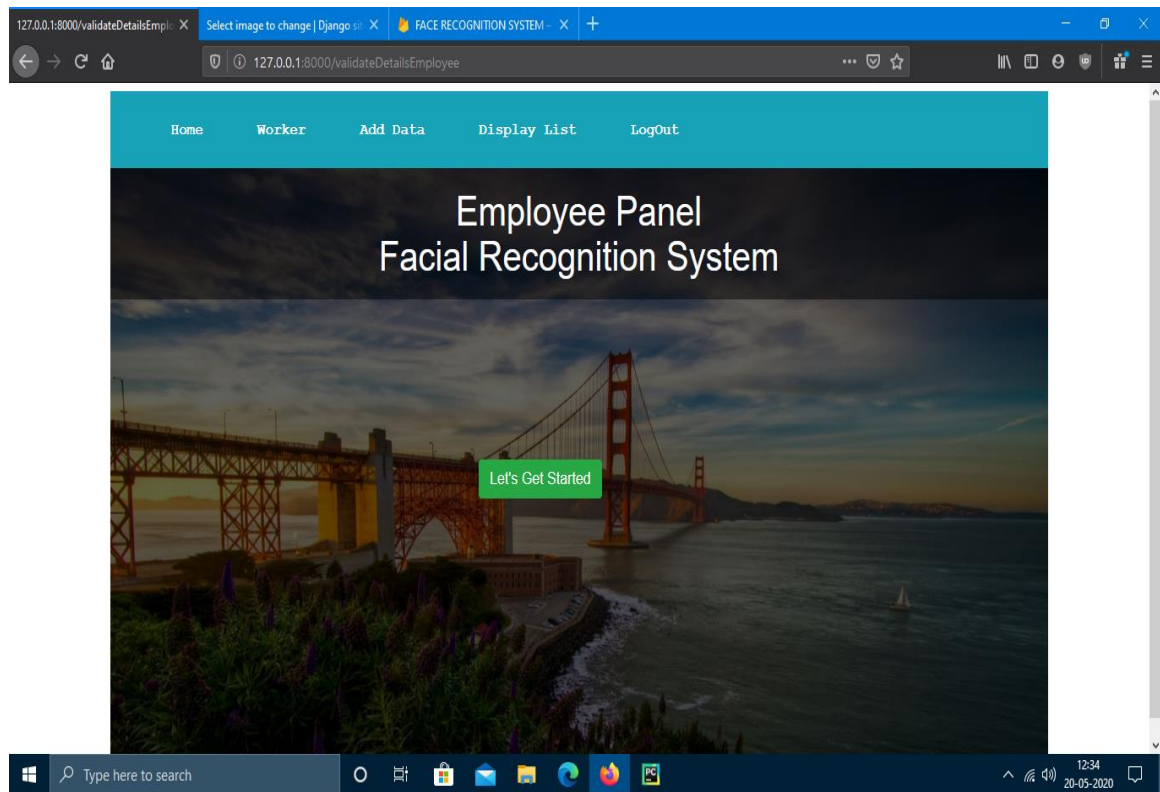


Figure 4 Home Page for Employee

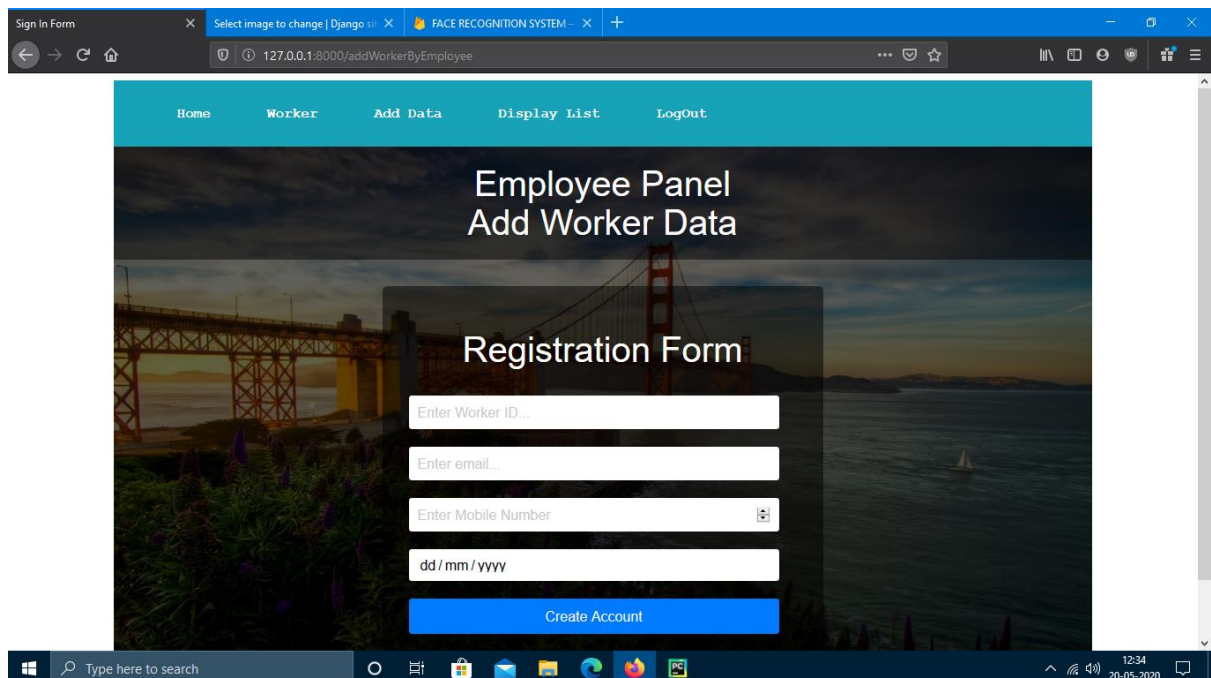


Figure 5 Add Worker by Employee

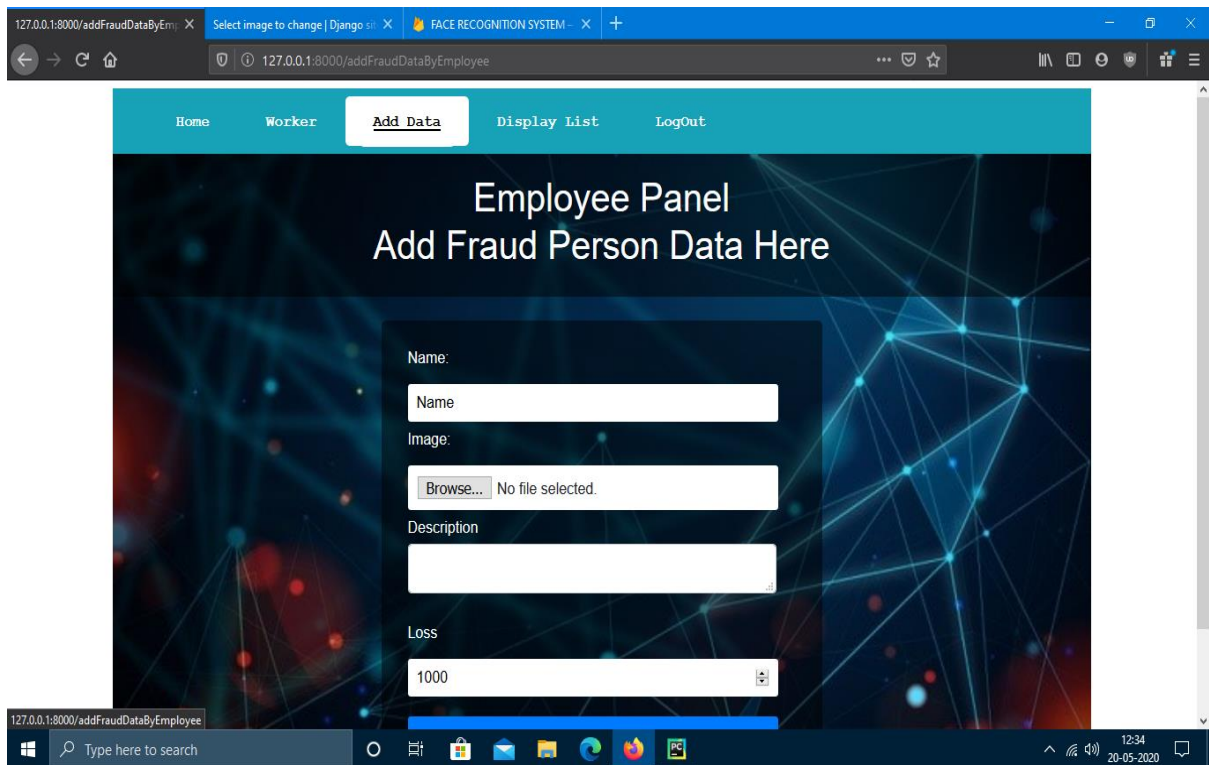


Figure 6 Add Fraud Person by Employee

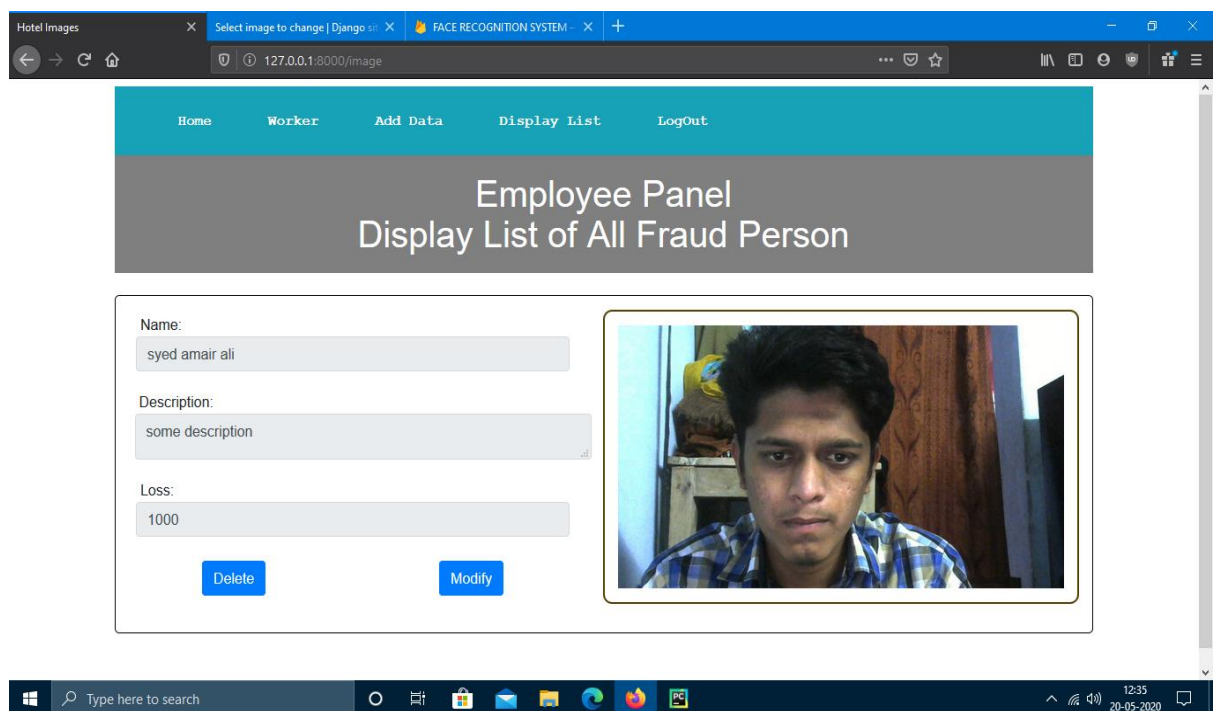


Figure 7 Display List View for Employee

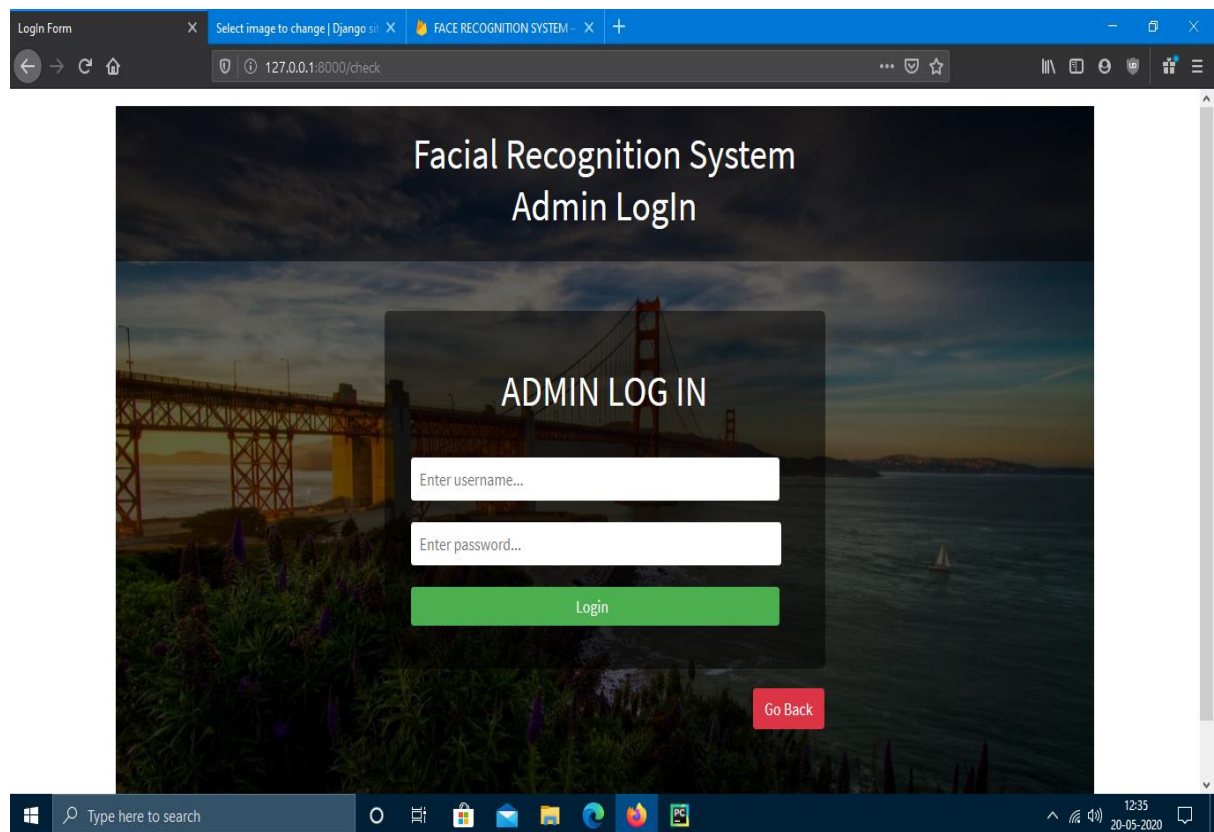


Figure 8 Admin Login Screen

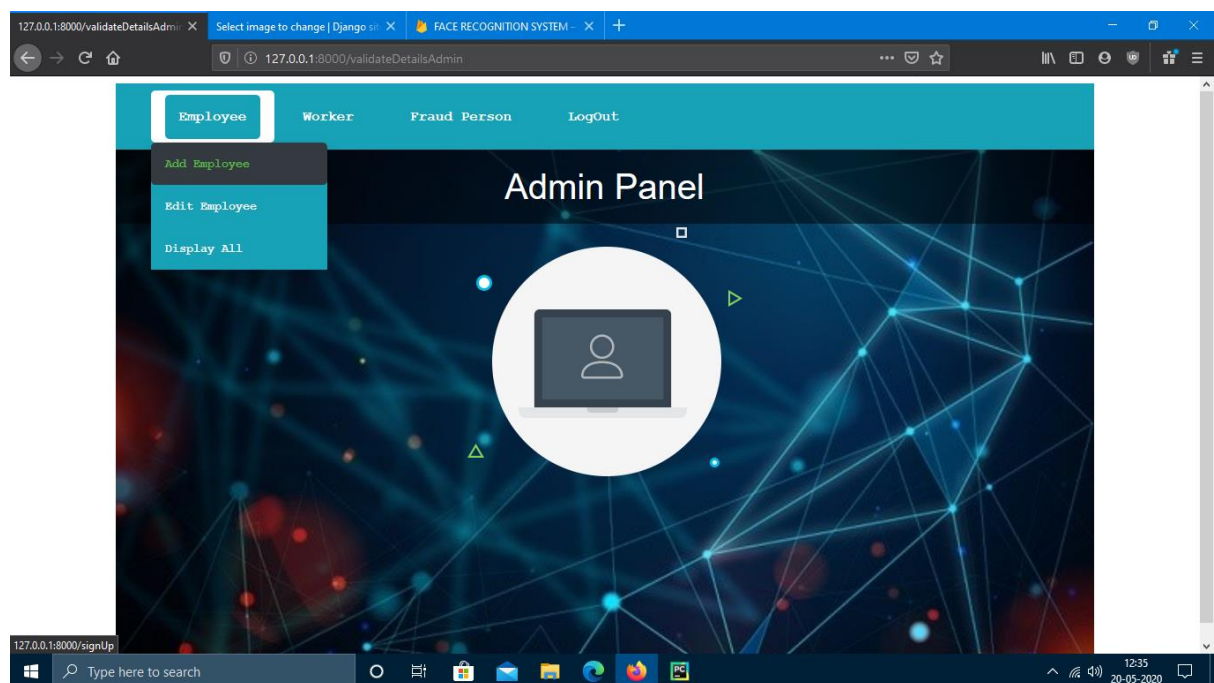


Figure 9 Admin Home Page

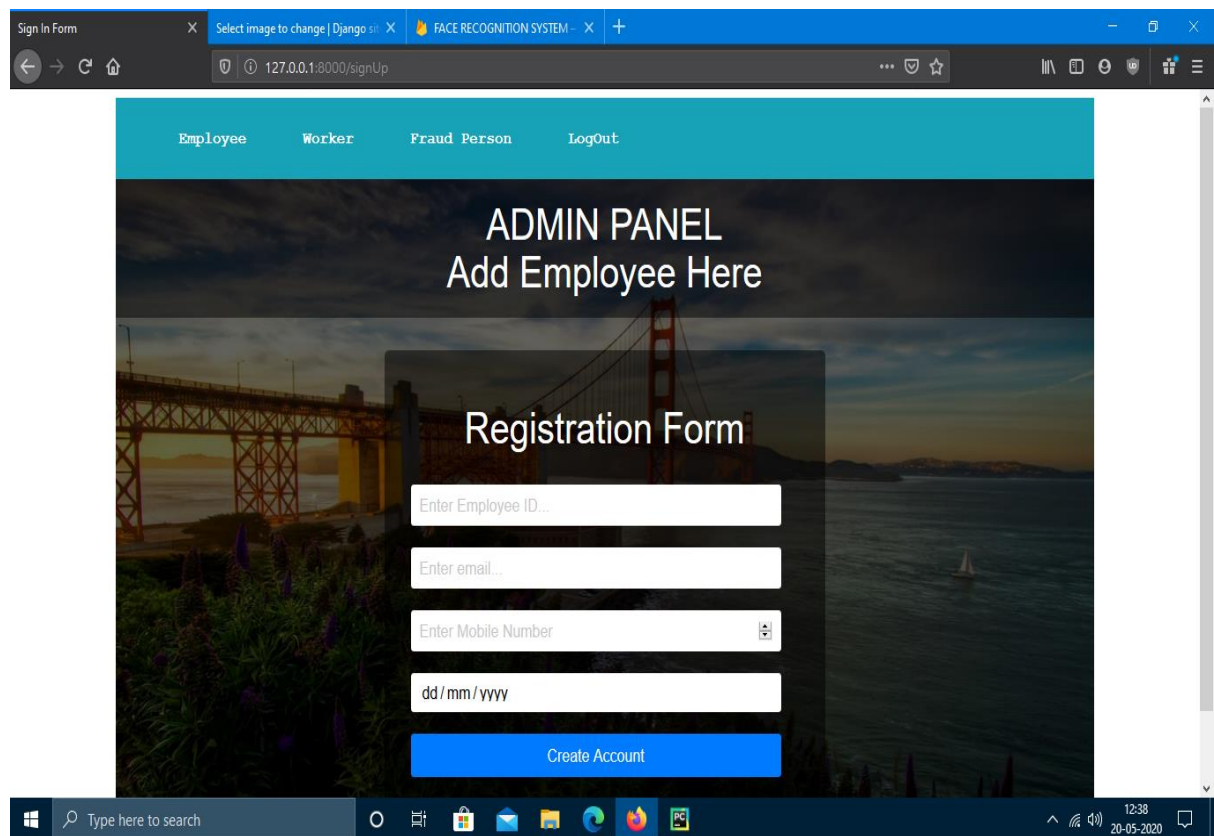


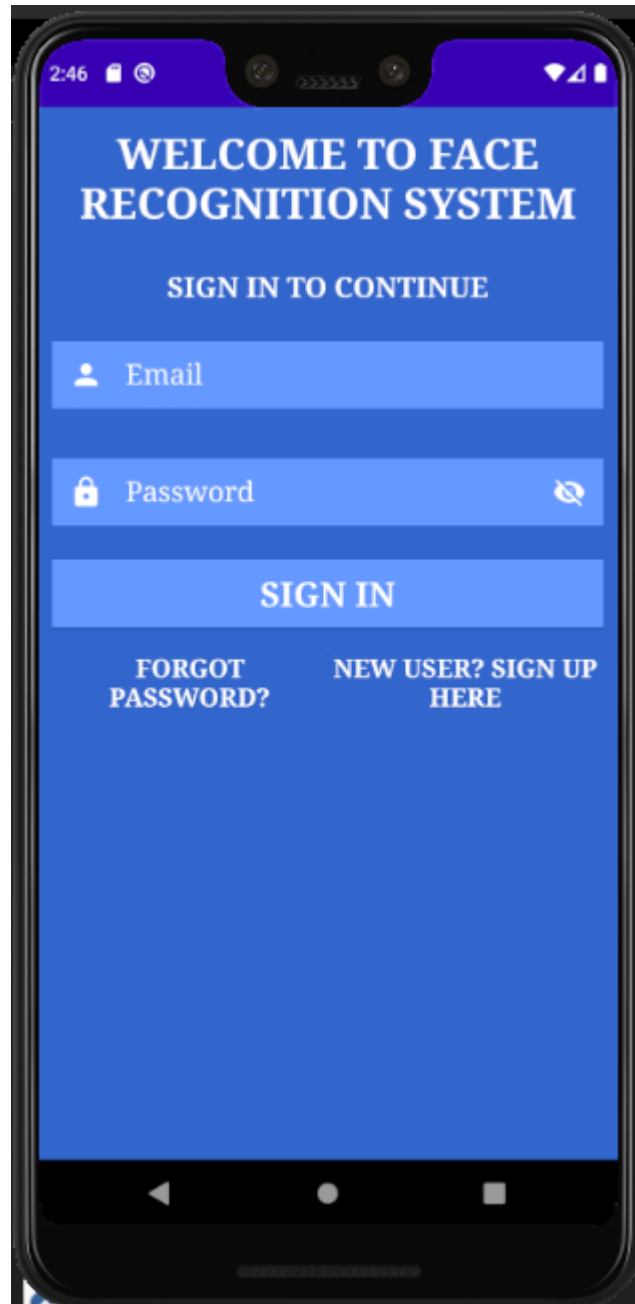
Figure 10 Add Employee by Admin



Figure 11 Display List for Admin



Welcome Screen



Log In Page for Worker

6:46

←

CREATE ACCOUNT 1 / 3

First Name
✎ MOHD SOHAIL ✕

Last Name
✎ AHMED ✕

Email
✉ [REDACTED]@gmail.com

Password
🔒 AzIsl) 👁

Confirm Password
🔒 AzIsl) 👁

For security, your password must be at least 8 characters. With at least 1 number, 1 special character and 1 uppercase letter.

GENERATE STRONG PASSWORD HERE

Creating Account Part 1

6:46

←

CREATE ACCOUNT 2 / 3

Male

Select Your Date-Of-Birth

Aug	11	1995
Sep	12	1996
Oct	13	1997

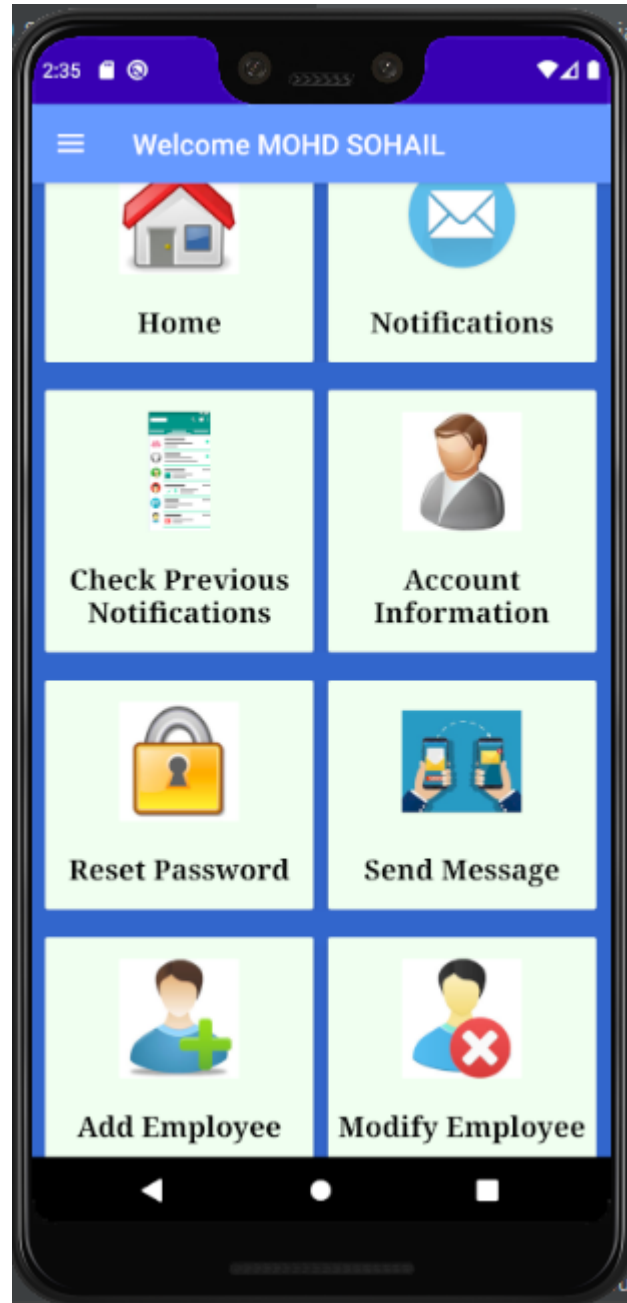
NEXT

LOGIN

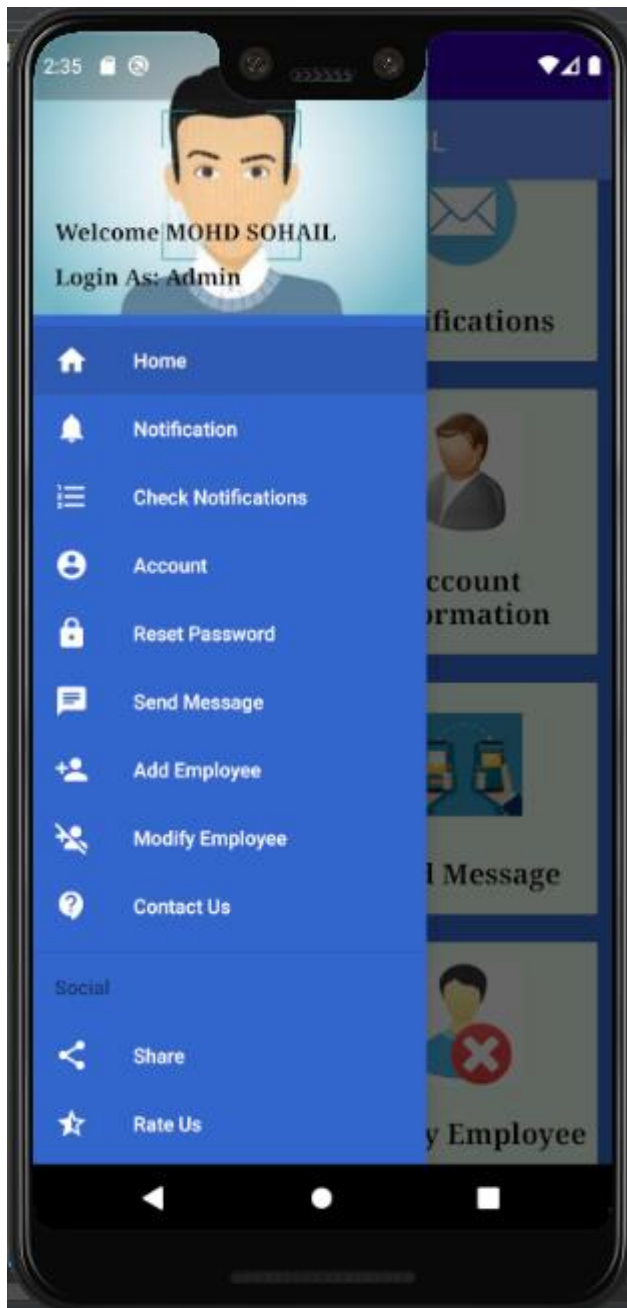
Creating Account Part 2



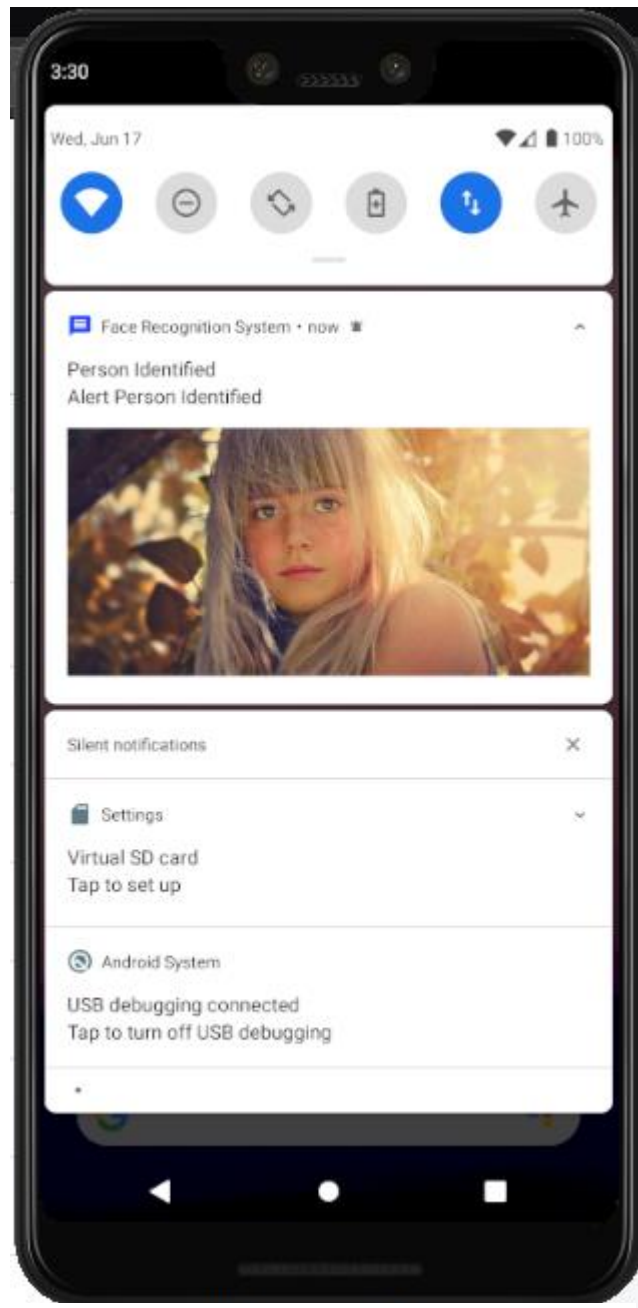
Creating Account Part 3



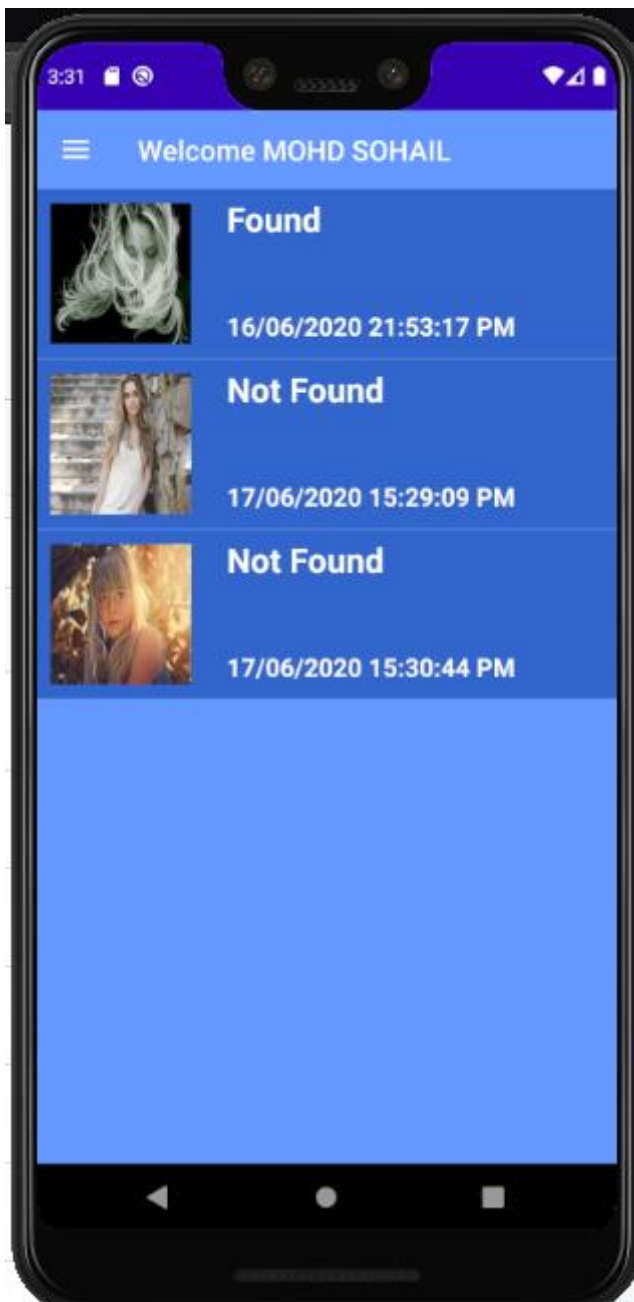
Home Page For Worker



Navigation Menu



Notification Panel



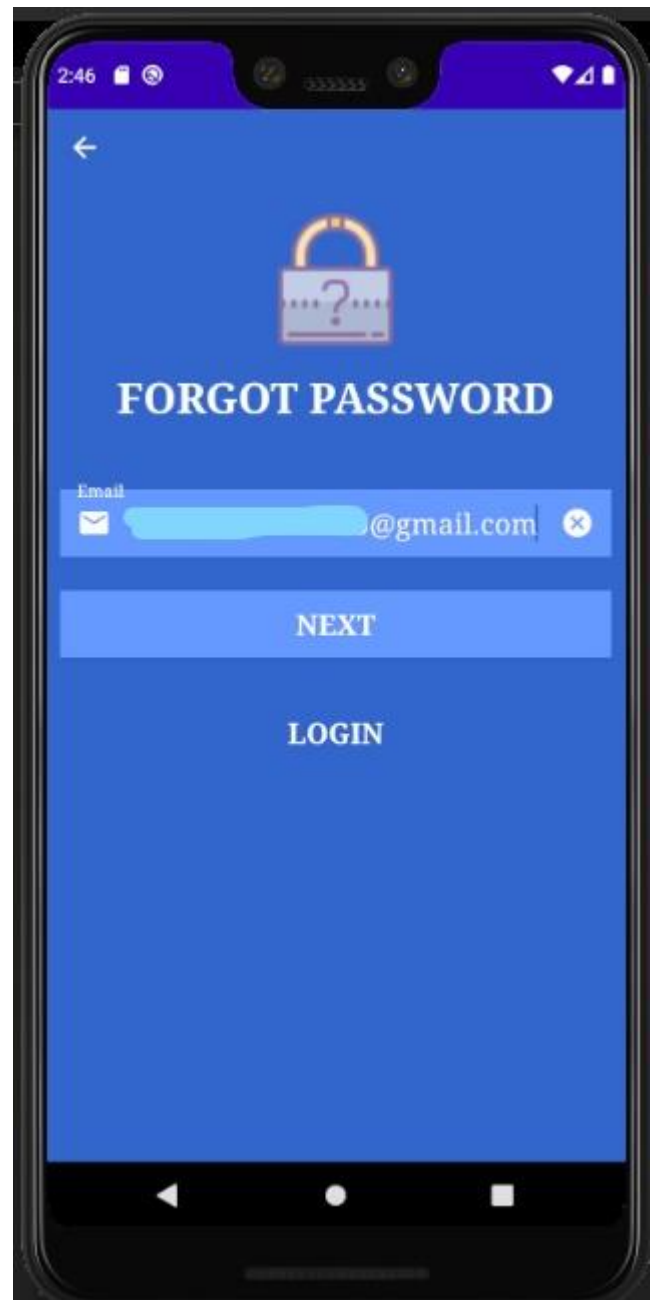
Previous Notification List



Detail View of a Particular Person



Current Worker Password Change Option



Forgot Password Page



Two Modes to receive OTP



Generation of Password

IV. Test Plans

24)Features to be Tested / Not to be Tested

a) Features to be Tested

The following are the features that are needed to test such as:

- I. User Login.
- II. Signup.
- III. Forgot Password.
- IV. Identifying the human face and recognizing that person.
- V. Sending Notifications on the user's android devices.
- VI. Displaying Notification's data to the users.
- VII. Data is saved into the database.

b) Features Not to be Tested

The following are the features that not to be tested such as:

- I. How notification is sent from the server side.
- II. How notification is handled on the client side.
- III. How our software identifies and recognizes faces.
- IV. How many notifications are sending per day?
- V. In which format notification data is sent.

25)Pass / Fail Criteria

I) User Login

When any user tries to login, the application should allow only authentic person's to login (i.e. who has a valid user name and password). Otherwise, the login test case fails and display an error.

II) Signup

When any new user tries to sign up, he/she should fill all the required fields in order create their accounts. If any required field is empty, the signup test case fails and display an error about which field is required.

III) Forgot Password

When any user of our application forgets their password, they should simply their email id in order to reset their password. This test case passes only when the users has created their accounts otherwise this test case fails and display an error that we do not recognize this email.

IV) Identify the human face and recognize that person

When our software sees any person, it captures its face and check for existing records in the storage. If match is found, it will notify its users about that person by sending data. To pass this test case, the face in the image or in video should be clear otherwise, it doesn't identify the person's and doesn't recognize that person. And for identification of that person, that person should not ware sunglasses or cover their face with mask or cloths. Due to such reason's the test case fails.

V) Sending notifications on the user's android devices

When our software identifies and recognizes any person, it sends a notification to its users. In notification, it sends the detail about that person such as image of that person, name, loss, description, and many more. To pass this test case, when our Django application sends notification, our Android users should receive the notification. Otherwise, test case fails.

VI) Displaying notifications data to the user

When the Android user receives notification from the Django application, the android application should display the data it's the users. when the application is in the background or application is killed, then user receive any notification and click on the notification, it will first check if the user is login or not if login, it will show data to the user. If the user is not login, the application tells the user to first login. And when the application is in foreground, it displays the notification data directly to the user. If all these conditions satisfied, test is considered to pass else the test case if failed.

VII) Data is saved into the database

When create an account, it is very important to save the user's data into the database for further usage. Also, when we train a person's model, it should also save the image as well as the details of that person into the database. If the data is saved into the database, this test case is satisfied otherwise, the test case fails.

26) Approach

In a software development process, testing is very important. By performing different types of testing, we can come across about potholes that are present in our software. Depending upon the module, we have performed different types of testing at different levels such as:

1. Unit testing
2. Integration testing
3. System testing
4. Acceptance testing

1) Unit testing:

In unit testing, we have performed testing in the individual modules such as login, signup, forgot password, face identification and recognition, sending notifications, displaying notifications and many more modules. The different types of testing we have user are:

- a. **Branch Testing:** This testing is used for testing each branch / part of our module / software.
- b. **Graphical User Interface (GUI) Testing:** This type of testing helps us to validate the gui is as per customer's requirements.
- c. **Gorilla Testing:** This type of testing helps us to test modules and there functionality thoroughly and heavily.

2) Integration Testing:

In integration testing, we perform testing by combining to or more modules together. The different types of integration testing we used are:

- a. **Component Testing:** This testing tells the behaviour of our software when we combine different modules together.
- b. **Integration Testing:** In this testing type, we combine two or more modules together to test their behaviour.
- c. **Incremental Integration Testing:** In this testing, the developers integrate the modules one by one using stubs and drivers to uncover defects in the software program.

3) System Testing:

System testing helps us to test the behaviour of our software when it communicates with database, network, other applications and hardware devices. The types of system testing we have used are:

- a. **Functional Testing:** This testing makes sure that the functionality of a product is working as per the requirements specification, within the capabilities of the system. Functional testing is done manually or with automated tools.
- b. **Recoverability Testing:** This testing determines whether operations can be continued after a disaster or after the integrity of the system has been lost.
- c. **Performance Testing:** This testing makes sure the system's performance under the various condition, in terms of performance characteristics.

4) Acceptance Testing:

In acceptance testing, we run the final test case for our software to check for bugs. Different type of acceptance testing we use are:

- a. **User Acceptance Testing:** This type of testing is used to test the product whether it satisfies the customer's demands and working correctly for the customers.
- b. **Operational Acceptance Testing:** This type of testing mainly includes testing of recovery, compatibility, maintainability, technical support availability, reliability, fail-over, localization etc.
- c. **End-to-End Testing:** This type of testing tells us the real-world use of our applications such as interacting with our database, network, other applications, or with hardware.

27) Suspension and Resumption

a) Suspension Criteria

When a tester is performing testing on a software, the process may halt due to certain conditions such as system crash, software failure, hardware, network failure, data is not received or stored into the database, and due to many more such reasons. Its developer's responsibility to find and fix bugs that are present in the software. After the fixing of bugs, the testing process can resume.

b) Resumption Requirements

When the developers fixed those bugs, the testers should test the software with some important testing types such as:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Monkey Testing
5. Compatibility Testing
6. Back-end Testing
7. End-to-End Testing
8. Operational Testing
9. Load Testing
10. Negative Testing

28) Testing Materials (Hardware / Software Requirements)

a) Hardware Requirements

The following are the hardware requirements for testing. They are:

1. Inter Core ith generation processor or any AMD processor.
2. 2GB RAM.
3. 2GB of free disk space for installation.
4. 256KB Cache memory for temporary files.
5. A display screen with 1280 x 1024 or higher screen resolution.
6. Mouse or other pointing devices.
7. A camera for surveillance.

b) Software Requirements

The following are the software requirements for testing. They are:

1. A 32-bit or 64-bit operating system like windows 7 or later version.
2. Any modern browser.
3. Internet connection.
4. Python and Django should be installed on the system with some libraries that are required in order to run the software.
5. An android device with API level 24+ or android version 7.0+.

29) Test Cases

27.1) Android Application Test Cases

27.1.1) Android Application Login

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	User login with valid user name and password when the user is connected to the internet.	1. Open the android application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Valid user Name and Password when connected to the internet.	U: user123@gmail.com P: user123	User should login into application and should display a welcome message.	Login Success	Pass
T02	User login with valid user name and password when the user is not connected to the internet.	1. Open the android application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Valid user Name and Password when not connected to the internet.	U: user123@gmail.com P: user123	User should not login into application and should display an appropriate error message.	Login Failed	Fail
T03	User login with invalid user name and	1. Open the android application. 2. Enter the User Name.	Invalid user name and/or password when	U: user123@gmail.com	User should not login into application and should	Login Failed because of	Fail

	password when connected to the internet.	3. Enter the Password 4. Click on login.	connected to the internet	P: admin123	display an appropriate error message.	invalid input	
T04	User login with invalid user name and password when the user is not connected to the internet.	1. Open the android application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Invalid user name and/or password when connected to the internet	U: user123@gmail.com P: admin123	User should not login into application and should display an appropriate error message.	Login Failed because the user is not connected to the internet	Fail
T05	User tries login with one or more fields empty.	1. Open the android application. 2. Click on login.	Empty user name and/or password fields.	U: P:	User should not login into application and should display an appropriate error message.	Login Fail because of no input data	Fail

27.1.2) Android Application Forgot Password

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pas s/Fa il
T01	User tries to reset his/her password with valid email address and when the user is connected to the internet.	1. Open the android application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Valid email address when connecte d to the internet.	E: user123@gmail.com	User should receive an email reset mail on their email address.	Passwor d reset link send successf ully	pass
T02	User tries to reset his/her password with valid email address and when the user is not connected to the internet.	1. Open the android application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Valid email address when the user is not connecte d to the internet.	E: user123@gmail.com	User should not receive an email reset mail on their email address and should get an error message.	Passwor d reset link should not send to the user because of invalid email address	Fail
T03	User tries to reset his/her password	1. Open the android application.	Invalid email address when the	E: admin123@gmail.com	User should not receive an email reset mail	Sending password reset link failed	Fail

	with invalid email address and when the user is connected to the internet.	2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	user is connected to the internet.		on their email address and should get an error message on the screen.	due to invalid email address	
T04	User tries to reset his/her password with invalid email address and when the user is not connected to the internet.	1. Open the android application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Invalid email address when the user is not connected to the internet.	E: admin123@gmail.com	User should not receive an email reset mail on their email address and should get an error message on the screen.	Sending password reset link failed due to no internet connection	Fail
T05	User tries to reset his/her password with an empty email address field.	1. Open the android application. 2. Click on send email button.	Empty email address.	E:	User should receive an error message on the screen.	Reset password failed because of no input data	Fail

27.1.3) Android Application Create Account

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	User tries to create an account with valid data and when the user is connected to the internet.	1. Open the android application. 2. Click on create account. 3. Filled all the required fields with valid data. 4. Click on signup button.	User has filled valid data and connected to the internet.	F: John L: Corner DOB:01/01/1990 G: Male PH:1234 567890 E: john123@gmail.com P: john123 CP: john123	Account should be created and user should redirect to the login page.	Account created and user is redirected to the login page	Pass
T02	User tries to create an account with valid data and when the user is not connected to the internet.	1. Open the android application. 2. Click on create account. 3. Filled all the required fields with valid data. 4. Click on signup button.	User has filled valid data and not connected to the internet.	F: John L: Corner DOB:01/01/1990 G: Male PH:1234 567890 E: john123@gmail.com	Account should not be created and user should not redirect to the login page.	Account creation failed and display an error message to the user and control stays on the same page	Fail

				P: john123 CP: john123			
T03	User tries to create an account with invalid data and when the user is connected to the internet.	1. Open the android application. 2. Click on create account. 3. Filled all the required fields with valid data. 4. Click on signup button.	User has filled invalid data and connected to the internet.	F: John L: Corner DOB:01/01/1990 G: Male PH:1234567890 E: john123g@mail.com P: john123 CP: john123	Account should not be created and user should not redirect to the login page.	Account creation failed and display an error message to the user and control stays on the same page	Fail
T04	User tries to create an account with invalid data and when the user is not connected to the internet.	1. Open the android application. 2. Click on create account. 3. Filled all the required fields with valid data. 4. Click on signup button.	User has filled valid data and not connected to the internet.	F: John L: Corner DOB:01/01/1990 G: Male PH:1234567890 E: john123g@mail.com P: john123	Account should not be created and user should not redirect to the login page.	Account creation failed and display an error message to the user and control stays on the same page	Fail

				CP: john123			
T05	User tries create an account with one or more fields empty.	1. Open the android application. 2. Click on create account. 3. Not filled all the required fields with valid data. 4. Click on signup button.	User has left one or more fields empty.	F: L: Corner DOB: G: Male PH:1234 567890 E: john123g mail.com P: CP: john123	Account should not be created and user should not redirect to the login page.	Account creation Failed because of no input data in one or more field	Fail

27.1.4) Android Notification When the Application is in Background

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	Notification is received when the application is in background or killed and when the user clicks on the notification, the data is displayed to the user when the user is logged in and connected to the internet.	Click on the received notification or open the android application to see the notification data.	Notification is received and data is displayed to the user.	Django application sends the data in the form of notifications to the android users.	When the user clicks on the notification or open application to see notification data, if user is logged in then controls goes to the home page and shows the notification data.	The user receives a notification when he/she clicks on the notification or open the application, the data is shown because the user is logged in	Pass
T02	Notification is received when the application is in background or killed	Click on the received notification or open the android application to see the	Notification data is not shown to the user	Django application sends the data in the form of notificati	When the user clicks on the notification or open application to see	The user receives a notification when he/she clicks on	Pass

	and when the user clicks on the notification, the data is displayed to the user when the user is not logged in and connected to the internet.	notification data.		ons to the android users.	notification data, if user is not logged in then controls goes to the login page and data is not shown to the user until the user is logged in.	the notificati on or open the applicati on, the data is not shown because the user is not logged in	
T03	Notification is not received when the application is in background or killed because the user is not connected to the internet.	Open the application to see notification data.	No data to show to the user.	Django applicati on sends the data in the form of notificati ons to the android users.	When the user opens the application to see notification data, if user is not connected to the internet, it will show a message to the user to connect to the internet to receive	If the user is not connecte d to the internet, it will display dummy data to the user.	Pass

					notifications		
--	--	--	--	--	---------------	--	--

27.1.5) Android Notification When the Application is in foreground

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	Notification is received when the application is in foreground when the user is connected to the internet.	Application should be opened.	Notification is received and data is received to the user when the application is in foreground.	Django application sends the data in the form of notifications to the android users.	When the application is in foreground and when user receives a notification, it shows an alert message about the notification to the user.	Notification is received when the application is in foreground and alert is shown to the user	Pass
T02	Notification is not received when the application is in foreground when the user is not connected to the internet.	Application should be opened.	Notification is not received and no data is available to show to the user when the application is in	Django application sends the data in the form of notifications to the android users.	When the application is in foreground and when user didn't receive a notification, it shows dummy data to the user because the	If the notification is not received, dummy data is shown to the user	Pass

			foreground.		user is not connected to the internet.		
T03	Notification is not received when the application is in foreground because the user is not connected to the internet.	Application should be opened.	Notification is not received and no data is available to show to the user when the application is in foreground.	Django application sends the data in the form of notifications to the android users.	When the application is in foreground and when user didn't receive a notification, it shows dummy data to the user because the user is not connected to the internet.	If the notification is not received, dummy data is shown to the user	Pass

27.2) Web Application Test Cases

27.2.1) Upload Data on the Server

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Pass/Fail
T01	Upload data to the Server	Text Field should be enabled	1. Enter Name 2. Select Image 3. Enter description 4. Enter Loss 5. Click on Upload button	N: Amair I: Image D: scent bottles damaged. L: 1400	Uploaded Successfully	Uploaded Successfully	Pass
T02	Upload data to the Server	Text Field should be enabled	1. Enter Name 2. Select Image 3. Enter description 4. Enter Loss 5. Click on Upload button	N: Sohail I: Image D: steeled some goods L: -1200	Uploaded Successfully	Loss must be Greater than Zero	Fail
T03	Upload data to the Server	Text Field should be enabled	1. Enter Name	N: I: Image	Uploaded Successfully	Name field cannot	Fail

			2. Select Image 3. Enter description 4. Enter Loss 5. Click on Upload button	D: some description L: 1234		be Empty	
T04	Upload data to the Server	Text Field should be enabled	1. Enter Name 2. Select Image 3. Enter description 4. Enter Loss 5. Click on Upload button	N: Omer I: Image D: description about omer L: 1234	Uploaded Successfully	Uploaded Successfully	Pass

27.2.2) Modify Data:

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Pass/Fail
T01	Modify the existing Data	All Text Fields should be enabled	1. Click Modify button 2. Modify required data 3. Click on Save Button	N: Amair Ali D: scent bottles damaged. L: 1400	Updated Successfully	Updated Successfully	Pass
T02	Modify the existing Data	All Text Fields should be enabled	1. Click Modify button 2. Modify required data 3. Click on Save Button	N: Sohail D: steeled some goods L: -1200	Updated Successfully	Loss must be Greater than Zero	Fail
T03	Modify the existing Data	All Text Fields should be enabled	1. Click Modify button 2. Modify required data 3. Click on Save Button	N: D: some description L: 1234	Updated Successfully	Name field cannot be Empty	Fail

T04	Modify the existing Data	All Text Fields should be enabled	1. Click Modify button 2. Modify required data 3. Click on Save Button	N: Omer D: description about omer L: 1234	Updated Successfully	Uploaded Successfully	Pass

27.2.3) Delete Data:

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Pass/Fail
T01	Deletes the existing Data	All Text Fields should be enabled	1. Click on Delete button	N/A	Item Deleted Successfully	Item Deleted Successfully	Pass
T02	Deletes the existing Data	All Text Fields should be enabled	1. Click on Delete button	N/A	Item Deleted Successfully	Item Deleted Successfully	Pass

27.2.4) Recognition:

Test Case ID	Test Case Objective	Prerequisite	Steps	Input Data	Expected Output	Actual Output	Pass/Fail
T01	To recognize the faces from live video stream	At least one entry of Data in the database	1. Click on Start button	If data is available	Recognize s faces and pushes notificatio n	Recogn izes faces and pushes notifica tion	Pass
T02	To recognize the faces from live video stream	At least one entry of Data in the database	1. Click on Start button	If data is not available	Recognize s faces and pushes notificatio n	This process requires data and Databas e is empty	Fail

27.2.5) Web Application Login:

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	User login with valid user name and password when the user is connected to the internet.	1. Open the Django application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Valid user Name and Password when connected to the internet.	U: user123@gmail.com P: user123	User should login into application and should display a welcome message.	Login Success	Pass
T02	User login with valid user name and password when the user is not connected to the internet.	1. Open the Django application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Valid user Name and Password when not connected to the internet.	U: user123@gmail.com P: user123	User should not login into application and should display an appropriate error message.	Login Failed	Pass
T03	User login with invalid user name and password when connected	1. Open the Django application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Invalid user name and/or password when connected to the internet	U: user123@gmail.com P: admin123	User should not login into application and should display an appropriate error message.	Login Failed because of invalid input	Pass

	to the internet.						
T04	User login with invalid user name and password when the user is not connected to the internet.	1. Open the Django application. 2. Enter the User Name. 3. Enter the Password 4. Click on login.	Invalid user name and/or password when connected to the internet	U: user123@gmail.com P: admin123	User should not login into application and should display an appropriate error message.	Login Failed because the user is not connected to the internet	Pass
T05	User tries login with one or more fields empty.	1. Open the Django application. 2. Click on login.	Empty user name and/or password fields.	U: P:	User should not login into application and should display an appropriate error message.	Login Fail because of no input data	Pass

27.2.6) Employee Forgot Password:

Test Case ID	Test Scenario	Test Steps	Test Data	Input Data	Expected Results	Actual Results	Pass/Fail
T01	User tries to reset his/her password with valid email address and when the user is connected to the internet.	1. Open the Django application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Valid email address when connected to the internet.	E: user123@gmail.com	User should receive an email reset mail on their email address.	Password reset link send successfully	pass
T02	User tries to reset his/her password with valid email address and when the user is not connected to the internet.	1. Open the Django application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Valid email address when the user is not connected to the internet.	E: user123@gmail.com	User should not receive an email reset mail on their email address and should get an error message.	Password reset link should not send to the user because of invalid email address	Fail
T03	User tries to reset his/her password	1. Open the Django application.	Invalid email address when the	E: admin123@gmail.com	User should not receive an email reset mail	Sending password reset	Fail

	with invalid email address and when the user is connected to the internet.	2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	user is connected to the internet.		on their email address and should get an error message on the screen.	link failed due to invalid email address	
T04	User tries to reset his/her password with invalid email address and when the user is not connected to the internet.	1. Open the Django application. 2. Click on forgot password. 3. Enter the email address. 4. Click on Send Email button.	Invalid email address when the user is not connected to the internet.	E: admin123@gmail.com	User should not receive an email reset mail on their email address and should get an error message on the screen.	Sending password reset link failed due to no internet connection	Fail
T05	User tries to reset his/her password with an empty email address field.	1. Open the Django application. 2. Click on send email button.	Empty email address.	E:	User should receive an error message on the screen.	Reset password failed because of no input data	Fail

30) Testing Schedule

Test Schedule ID:	TS01		
Product ID/Name:	Facial Recognition System		
Product Version:	1.1.0		
Present Owner/Created By:	MOHD SOHAIL AHMED, SYED AMAIR ALI and SHAIK ABDUL FARAZ		
Created on:	25-03-2020		
Updated on:	20-04-2020		
Last Updated on:	15-05-2020		
Updated By:	MOHD SOHAIL AHMED, SYED AMAIR ALI and SHAIK ABDUL FARAZ		
Host Name:	MOHD SOHAIL AHMED, SYED AMAIR ALI and SHAIK ABDUL FARAZ		
Runs on:	30 days		
Current Status Enabled:	No		
Linked Items:	<p>The following items are linked to the document:</p> <ul style="list-style-type: none"> • Project Description • Requirements • Design • Test Plans • Project Issues 		
Review Comments:	On last review, we have added some more test cases and added detailed information about our future enhancement plans of our project.		
Current Version:	1.1.0		
Change Details:	In last review of documentation, we focused on test cases of section-4 and section-5 in which we have mentioned detail about future enhancement of our project.		
Signing Off Authority:	Name	Position	Signature
	Mohd Sohail Ahmed	Android Developer	Sohail
	Syed Amair Ali	Web Application Developer	Amair
	Shaik Abdul Faraz	Testing	Faraz

V. Project Issues

31)Open Issues

The main issue of the existing face recognition system is that they take much time to train a model and require more than 300 images of a person to train a model. This process is slow. Another problem is that the existing system retrains the model whenever we want to train a model for a new person. The main drawback of the current facial recognition system is that the more images we give at the time of training the model, the higher accuracy we get at the time of identifying a person.

To overcome the existing system problems, we made software for identifying the person with higher accuracy and is very fast in comparison with traditional facial recognition systems. Our software identifies a person with more than 90% accuracy whereas, the existing piece of software gives 80% accuracy to identify a person. The main advantage of our software over the traditional facial recognition system is that it takes only one image of a person to train the model. To train a new person's image, it uses the previously trained model instead of training a new model. A new key feature of our software is that, when it identifies a person, it sends a notification to the workers of that place on their android devices. In notification, it contains the details of identified person like his/her photograph, name, how much loss that person has done, description and many more.

We intend to make software that has to overcome the existing system problems and to provide business like shopping malls, department stores, and much more software that helps those people to identify those people that had damage to the property or had damage to their goods in the past. The camera is placed on the entrance of these places and it will monitor all the people who are entering those places. When any person that damaged to this place's visits again, our software will identify that person and inform the workers about that person.

32)Off-the-Shelf Solutions

30.1) Ready-Made Products

When the time of building our facial recognition system software, we used an existing neural network model to train a person's image which is fast in comparison to this type of existing software. The main reason behind using this model is this model trains models very fast and has higher accuracy when identifying a person and it requires a single image of a person to train a model. Due to this reason, we have decided to use this model in our software.

30.2) Reusable Components

In our software, many things are reusable such as the model that we have used to train a person's image. We have created a notification channel for sending the notifications to the people who are working in that place. Which is a reusable component and there is no limit in sending notifications to the android devices. The android devices are also capable to store the previous notifications and the newly arrived notifications as well. Once the user is logged in in our android application, it doesn't ask them to log in again when the next time they visit this application until the user has logged out by them self's.

30.3) Products That Can Be Copied

To overcome the problems that existing facial recognition system has, we found a solution to an existing training model that helps us to train a person's image with the help of a single image and this model also identifies a person face with more than 90% accuracy which is considered a good one for recognizing human faces. This model is available openly for everyone and nowadays, many people are using this model in the projects because of its reliability, its accuracy, and many more things.

33)New Problems

31.1) Effects on the Current Environment

The traditional facial recognition system requires more than 300 images of a person to get more than 80% accuracy while identifying a person. And the training the model has to retrain from starting when training a new person's image which is a time-consuming process. In our software, we overcome these problems of the existing system. We have used a training model that requires a single image to train a model and it doesn't require to retrain the model for whenever training a new person's image and it has more than 90% in recognizing a person. As everything has some pros and cons, our software is also having some drawbacks. It cannot identify a person who covers their face and it gives less accuracy to recognize those persons who wear sunglasses.

31.2) Effects on the Installed Systems

Many existing facial recognition systems have an attractive interface but don't meet the requirements of the users who are using this software. While some are lag behind in functionality and some are lag behind in the behaviour. The software which we made meets the requirements of its users. We made our software so simple that everyone can use it with ease. It also instructs its users at each step when they need help. Moreover, we have added some new functionalities like informing the people who are working that place about the recognized person with that person's information such as an image of that person, name, description about the loss, and how much is the loss and many more which the existing system doesn't provide to its users.

31.3) Potential User Problems

The existing facial recognition system has many drawbacks such as it requires more than 300 images and it takes time to train a person's image. Moreover, when we want to new a person's image, it retains the entire model which is a time-consuming process and when it recognizes any person, it doesn't provide an interface through which it informs the people who are working in that place. Also, this model recognizes any person not more than 85% accuracy. This are some issues that users of existing facial recognition system users are facing.

31.4) Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

As everything has some pros and cons, our software also has some drawbacks such as it doesn't recognize the faces of those people who cover their faces by wearing a mask, or with the help of some cloth. When any person wears sunglasses, it recognizes that person with less accuracy. It is very difficult to identify the face or train the image when the face in the picture is not clear.

31.5) Follow-Up Problems

Our software requires a 24/7 electricity supply to work with ease. It also requires a good internet connection for our software as well as for our android devices to function properly. Till now, we didn't come up with a solution to recognize those people who cover their faces with masks or with some cloths. And also, to identify those people whose face in the picture is not clear. Maybe in future, we may find a better solution to overcome with these problems.

34)Tasks

32.1) Project Planning:

In the process of developing software, the developers have to follow the following six phases of the software development life cycle model (SDLC). This phase of SDLC helps developers to make software that should meet the customer's needs and satisfy customers. The following are the six phases of SDLC model.

- Requirement Gathering
- Analysis
- Design
- Coding
- Testing
- Development

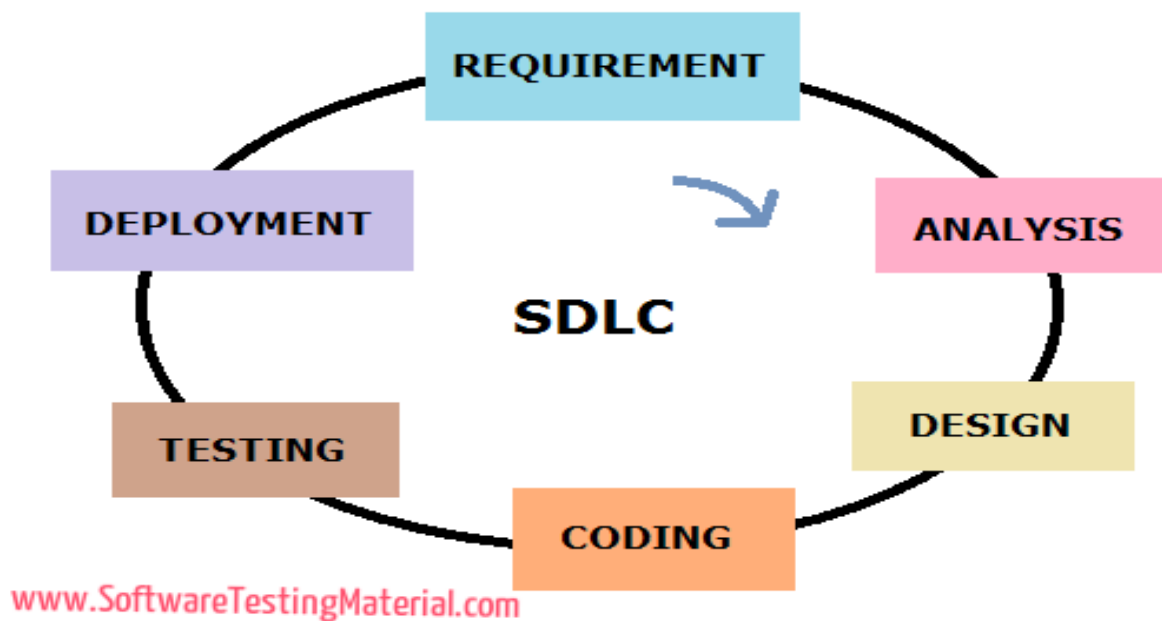


Figure-32.1: Software Development Life Cycle

1. Requirement Gathering:

In this phase, we gathered information about the existing system of facial recognition systems. During the requirement gathering phase, we came across some flaws that are present in the current facial recognition system which we tried to overcome in our project. We also came across various questions like who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirement gathering phase.

2. Analysis:

After gathering the requirements from the requirement gathering phase, we analysed the gathered information for the validity and the possibility of incorporating the requirements in the system. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3. Design:

In this phase, the system and software designs are prepared from the requirement specifications which we gathered in the requirement gathering phase. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture. And the software design helps us to understand the behaviour of our software and how the workflows from one activity to another activity.

4. Coding:

In this phase, we practically implemented our knowledge that we gathered from the requirement gathering phase. Based on the information, we divided our projects into modules/units. Since in this phase, we have to implement our work practically, it mainly focuses on the developer. This is the longest phase of the software development life cycle.

5. Testing:

After the code is done, the next step is to test our software to check whether its functionality is the same as we expected. It also helps us to make sure that all the requirements that gathered are fully satisfied. During the testing phase, we have performed some most commonly used and most important testing such as unit testing, integration testing, system testing, and acceptance testing.

6. Deployment:

After successfully testing our software under different testing conditions, we deployed our software in the current working environment. After the deployment of software, we again tested our software behaviour in the real working environment to check for any vulnerabilities.

32.2) Planning of the Development Phases

In the process of development on any software, proper planning and development of software in the current environment are the essential phases. With proper planning, we can develop software with ease.

1. Planning:

The first phase in the process of developing software in the planning phase. In this phase, we gathered information about the existing facial recognition systems. During the requirement gathering phase, we came across some flaws that are present in the current facial recognition system such as it requires more than 300 images to train a model when we train another person's image, it retains the model which is a time-consuming process, and many more which we tried to overcome in our project. We also came across various questions like who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? and many more. We also gathered information about the hardware and software requirements.

2. Analysis:

After gathering the requirements from the requirement gathering phase, we analysed the gathered information. After analysing the gathered information, we then wrote a rough draft and divided then modules. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3. Design:

The design phase is one of the important phases in the software development. In this phase, we make designs for our software, helps us to understand the behaviour of our software, how the workflows from one activity to another activity, and many more things.

4. Coding:

In this phase, we practically implemented our knowledge that we gathered from the requirement gathering phase. Based on the information, we divided our projects into modules/units. Since in this phase, we have to implement our work practically, it mainly focuses on the development. This is the longest phase of the software development life cycle model.

5. Testing:

After the coding is done, the next step is to test our software to check whether its functionality is the same as we expected. Testing phase helps us to make sure that all the requirements that gathered are fully satisfied. During this phase, we have performed some most commonly used and most important testing techniques such as unit testing, integration testing, system testing, and acceptance testing.

6. Deployment:

After successfully testing our software under different testing conditions, we deployed our software in the current working environment. After the deployment of software, we again tested our software's behaviour in the real working environment to check for any vulnerabilities and checked for any improvements that can be done.

35) Migration to the New Product

33.1) Requirements for Migration to the New Product

The traditional facial recognition systems lack behind in some of the features such as its process time for training a model is high because it takes more than 300 images of a person to train a model and it retains the entire model when we train a new person's image to our model. Due to these reasons, the processing time of the traditional face recognition system is very high. Another drawback is that, it gives 80-85% accuracy while recognizing the faces of the known person which is quite low. Lastly, if the system or software recognizes the face of any known person, it will not inform the workers about the identified person. Due to such reasons, we have decided to migrate to the new product of the facial recognition system in which we have solved all the above-mentioned problems.

33.2) Data That Has to Be Modified or Translated for the New System

In the traditional facial recognition system, most of the developers used to prefer the SQL database to store the data and a directory to store the images of the fraud persons. The SQL database is quite slow in retrieving data from it and storing data into it. And they also use a separate directory for each person to store the images in it which takes huge memory and, in the future, it may result in memory-related issues.

The solution to the above problem is to use a NoSQL database to store the data in it. It is fast in comparison to the SQL database and we can store the link of the place where the image is located. Data retrieval and data storage is fast when compared to the SQL database.

36)Risks

As every project or software has some risk, our facial recognition system also has some risk. The risks involved in our software are

- Our Django application requires an internet connection to send the notifications about the identified people to the workers on their android applications. If the internet connection is lost due to some reason, our Django application can't send notifications to its users.
- Our android applications require an internet connection as well in order to receive notifications and display notification data to the user. If the android devices are not connected to the internet, the users will not receive notifications send by our Django application.
- Another risk is that, the surveillance cameras will stop working due to power cut-offs or because they are damaged. In such conditions, our software can't recognize the faces of the people who are visiting those places.
- Another risk is that, our software can't recognize the faces of the people and can't send the notifications to its android users because due to come software failure or hardware failure.

37)Costs

The “Facial Recognition System” software which we build requires less cost to build it. In hardware, it requires only a camera for surveillance purposes. It also requires it requires a smartphone to receive notifications and display notification data to the used send by Django application. And lastly, internet connection 24/7 for the camera for surveillance purposes and for android devices to receive and display notification data.

38)Waiting Room

Our software is made using a neural network model to train a person's image. Though this model has some drawbacks such as it cannot recognize the people who cover their face with a mask or cloth or with anything. It also gives less accuracy while recognizing the people who wear sunglasses. These are some of the weak sides of our software. In the future, it may be possible to overcome this problem. The solution we thought of this problem is iris recognition. With iris recognition, it is possible to identify the people who cover their faces or hide their faces from others. And the solution to the next problem is to use feature analysis on the face. By using feature analysis, we can identify those people who wear goggles and sunglasses. These are the solutions to the problems that we have in our software. In the future, it is possible to overcome those problems by using these solutions.

39)Ideas for Solutions

The traditional facial recognition system has few drawbacks such as it requires more than 300 images of a person to train a model and it is a very time-consuming process to train those 300 images. Moreover, it gives 80-85% accuracy to recognize the face of a known person. And it retrains the entire model when training a new person's image which is a time-consuming process. So, we came to a solution to this problem. The solution is to use a neural network model to train a person's image. The advantages of this model are it requires a single image of a person to train the model. It requires a single image to train the model, so the process of training an image is very fast. It gives more than 90% accuracy to recognize the face of a known person. Moreover, it does not retrain the entire model while training a new person's image. Due to this reason, we have decided to use the neural network model instead of using the traditional face recognition model.

40)Project Retrospective

The neural network model which we have used in our project “Facial Recognition System” to train a person’s image was the effective model and have high accuracy when compared to the traditional face recognition model and it requires a single image to train a model whereas the traditional system requires more than 100 images of a person to train a model. The main disadvantage of this model is that it cannot recognize a person who wears a mask or cover their face. This model gives less accuracy in recognizing the faces of those people who wear sunglasses. These are some of the advantages and disadvantages of using the neural network model. Maybe in the future, we can come across a better solution to overcome the disadvantages of this model.

41)Conclusion

Fraud detection at shopping malls using facial recognition was designed and implemented. It was tested with different face images. Under controlled condition, when lighting and pose can be restricted, the system can recognize persons faces with more than 90% accuracy.

It is implemented using Django and Python which met the design criteria and solves the problem.

42)Future Enhancements

It includes following:

- To make algorithm more robust so that it can identify person with the masks on their face
- Recognizing algorithm with sun glasses on the face of the persons with high accuracy
- To make algorithm more scalable, so that we can get more accuracy even for the images with low quality
- To make the algorithm recognize the persons in the dark lighting conditions as well.

43) References

- <https://github.com/ipazc/mtcnn>
- <https://anaconda.org/anaconda/python>
- <https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/>
- <https://pusher.com/tutorials/push-notifications-android>
- <https://blog.datawow.io/face-detection-haar-cascade-vs-mtcnn-13af4aa180e6>
- <https://github.com/nyoki-mtl/keras-facenet>