PROJECT REPORT ON

Automated and No-Touch Attendance System

Team Members:
Bhanvi Dewan (18103014)
Abhishek Chauhan (18103029)
Harmanpreet Kaur (18103037)
Arzoo Goyal (18103087)

Project Mentors:
Prof. Poonam Saini,
Prof. Amandeep Kaur,
(Computer Science and
Engineering Faculty)

TABLE OF CONTENTS

INTRODUCTION

Our project aims to develop a system wherein a face recognition module will identify a person and then mark his/her attendance in the respective organization without using traditional methods like register entry or fingerprint-based biometric attendance, which are tedious and time-consuming. This will reduce the amount of paper resources needed in the management of attendance data while gauging and eliminating proxies. This could be even more beneficial in the recent pandemic period (Covid-19). A major benefit of this will be the eradication of the need to compile attendance manually, thus reducing the amount of administrative work.

The human face is a unique representation of individual identity. Face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time captured images with stored images of that person in the database.

Motivation

Our project aims to develop a system wherein a face recognition module will identify a person and then help to mark his/her attendance in the respective organization without using traditional methods like register entry or thumb or fingerprint-based biometric attendance. This could be even more beneficial in the recent pandemic period (Covid-19).

Hence, we have proposed an automated attendance system based on face recognition with the following benefits:

- Eliminate the time and effort wasted in taking attendance manually in organizations.
- Reducing the amount of paper resources needed in attendance data management.
- Gauging and eliminating proxies.
- Not having to compile attendance manually, instead store it in a database that can be accessed by authorized personnel only.
- Reduce the amount of administrative work.

Aims and Objectives

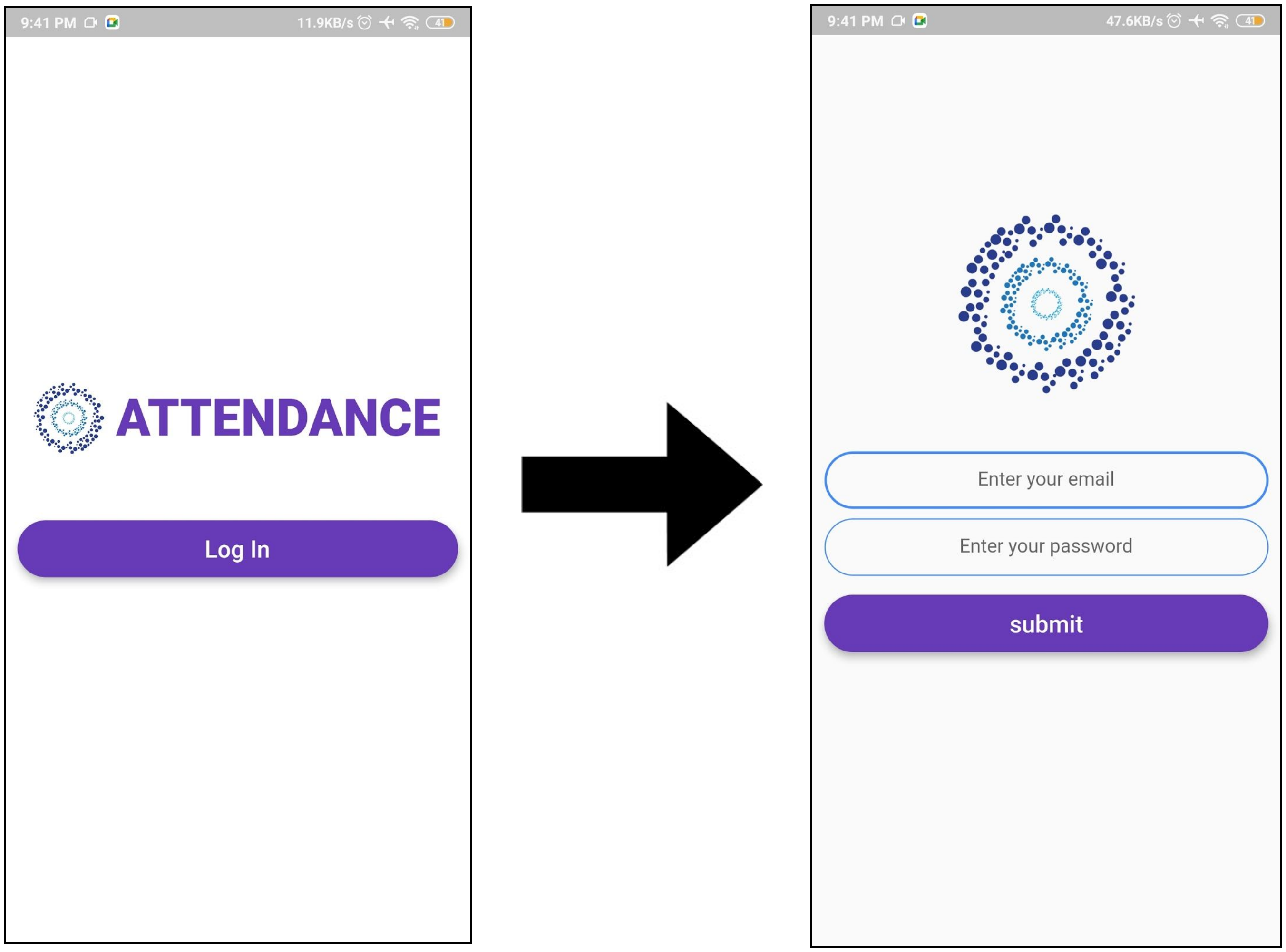
The objective of our project is to develop face recognition based automated attendance system with following tasks:

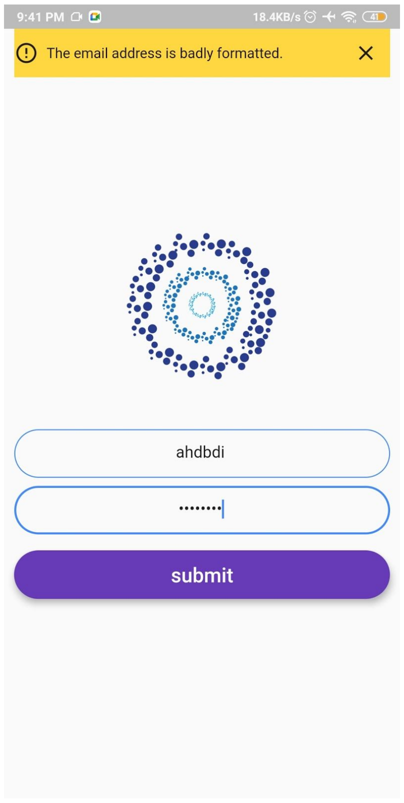
- To detect the face segment from the image.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the User.
- To notify each user that his/her attendance has been marked.
- To present a complete record of the user's attendance, including leaves, percentage, etc.

1. APP EXECUTION

1.1 LOGIN SCREEN

- When opening the Application for the first time, the user is shown the first screen for login process.

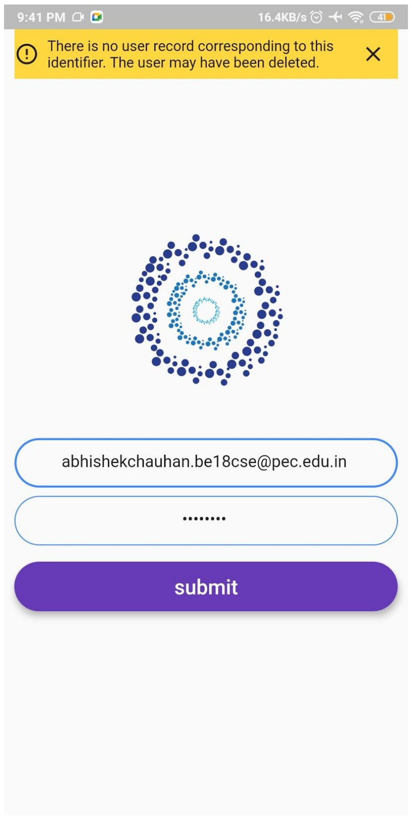



9:41 PM 18.4KB/s

The email address is badly formatted.

ahdbdi

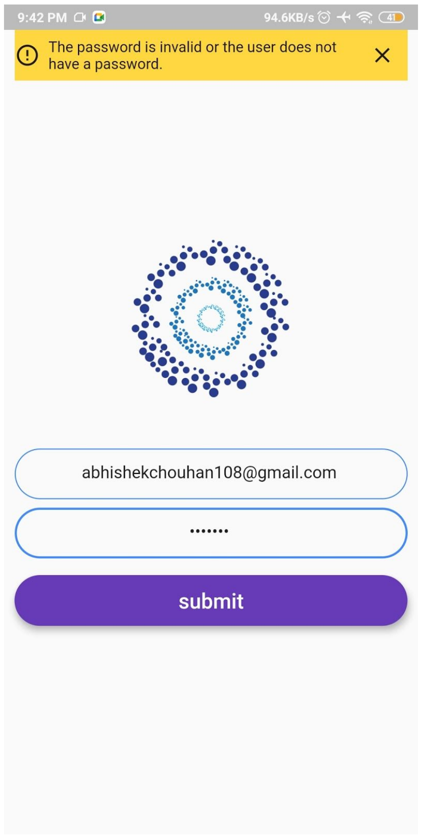
••••••••


9:41 PM 16.4KB/s

There is no user record corresponding to this identifier. The user may have been deleted.

abhishekchauhan.be18cse@pec.edu.in

••••••••


9:42 PM 94.6KB/s

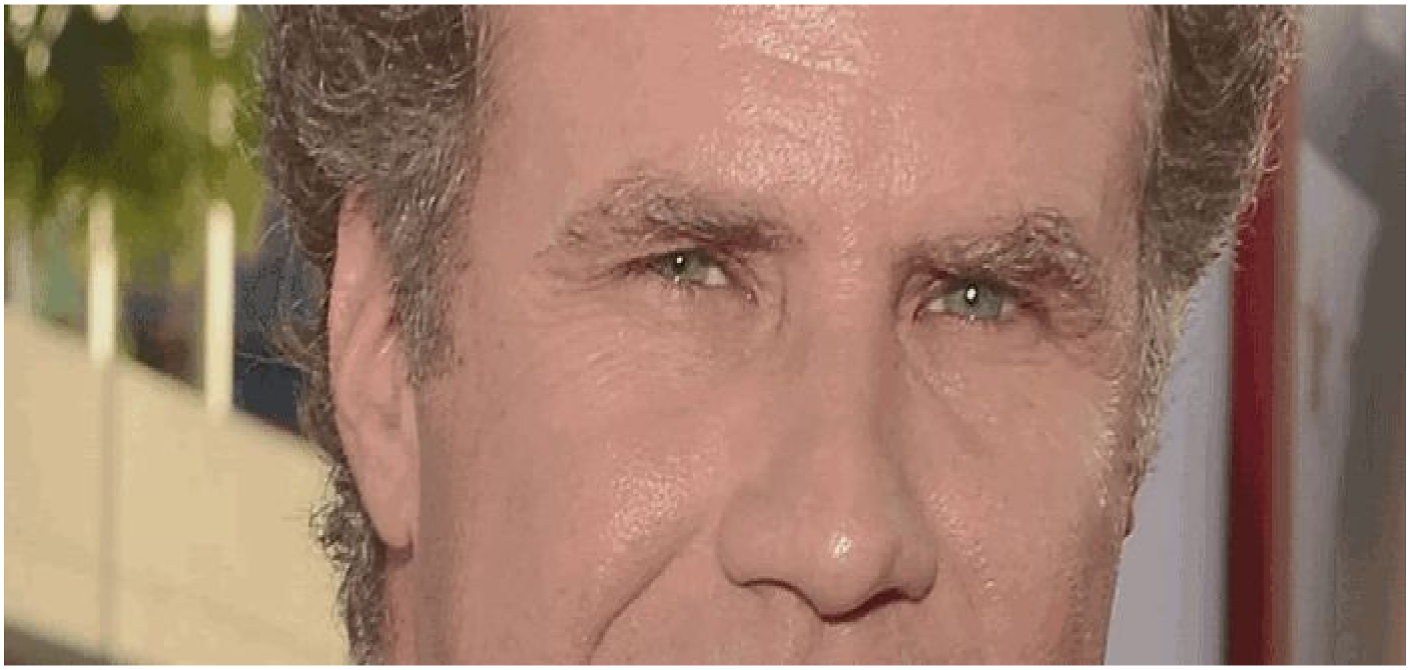
The password is invalid or the user does not have a password.

abhishekchouhan108@gmail.com

•••••••

Incorporation of Artificial Intelligence:

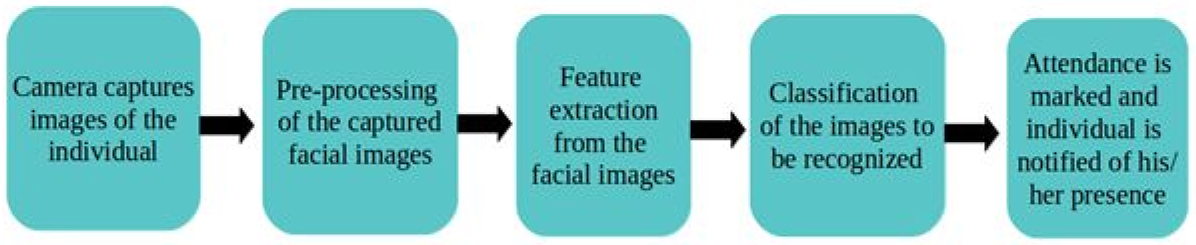
Artificial intelligence has been used for face recognition. We're using a library (named face-recognition) that uses a method called Histogram of Oriented Gradients — or just *HOG* for short. It starts by making the image black and white because we don't need color data to find faces. The system figures out how dark the current pixel is compared to the pixels directly surrounding it. Then it draws an arrow showing in which direction the image is getting darker. Repeating that process for every single pixel in the image, we end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image. But we do not need the data for every single pixel, so we'll break up the image into small squares of 16x16 pixels each. In each square, it'll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc.). Then it'll replace that square in the image with the arrow directions that were the strongest. The end result is that it turns the original image into a very simple representation that captures the basic structure of a face regardless of image brightness.



Next, it will try to warp each picture so that the eyes and lips are always in the sample place in the image. To do this, it uses an algorithm called face landmark estimation. It comes up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then it will train a machine learning algorithm to be able to find these 68 specific points on any face. It'll simply rotate, scale and shear the image so that the eyes and mouth are centred as best as possible. It is only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations). But we also need to be able to recognize faces in milliseconds, not hours. What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. The solution is to train the model to generate 128 measurements for each face.

The model learns to reliably generate 128 measurements for each person. The 128 measurements of each face are called an embedding. Now it finds the person in our database of known people who have the closest measurements to our test image. It uses a simple linear SVM classifier.

Work-Flow


Camera captures images of the individual
Pre-processing of the captured facial images
Feature extraction from the facial images
Classification of the images to be recognized
Attendance is marked and individual is notified of his/ her presence

Incorporation of Web Technologies:

The technologies which we have utilized are Google Firebase and Flutter.
Flutter in Android Studio makes up our front end. The app has been written entirely in Dart (a language in the Flutter package).

Google Firebase is the server for the project, of which we have utilized the following services:

- Firebase Authentication is employed at the login-in stage to authenticate the user.

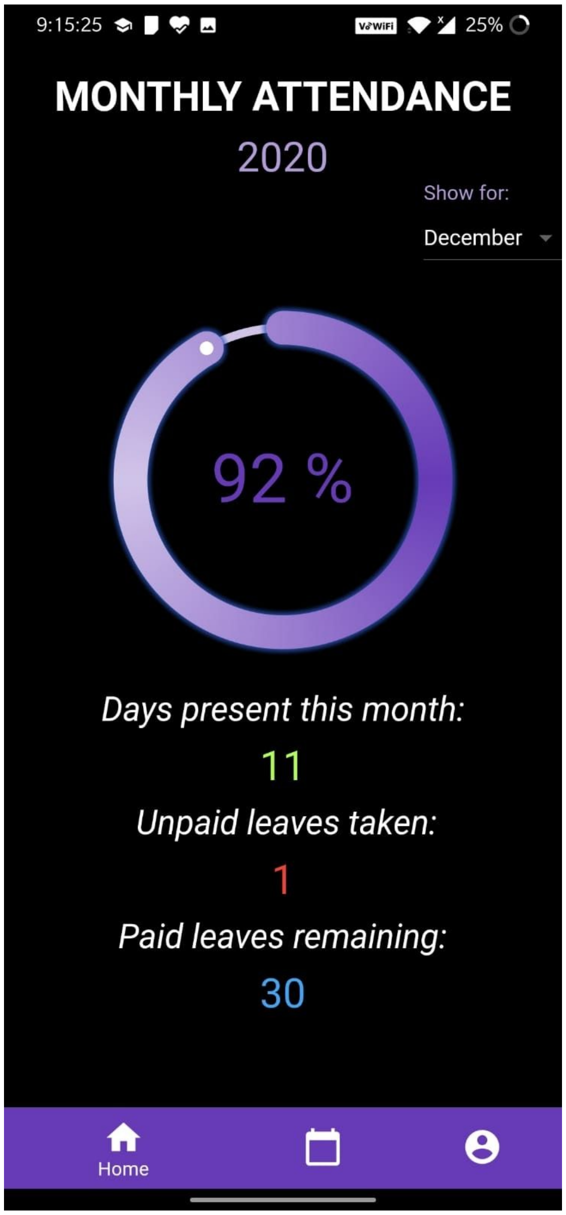
- We have used Cloud Firestore, a NoSQL document-oriented database in which the structures alternate between documents and collections of documents. This contains the entire record of the attendance. The entire database is created and the data updated using python. Cloud Firestore is accessed by this code that marks attendance using face recognition, and also by the app, that reads and displays the data to the user.

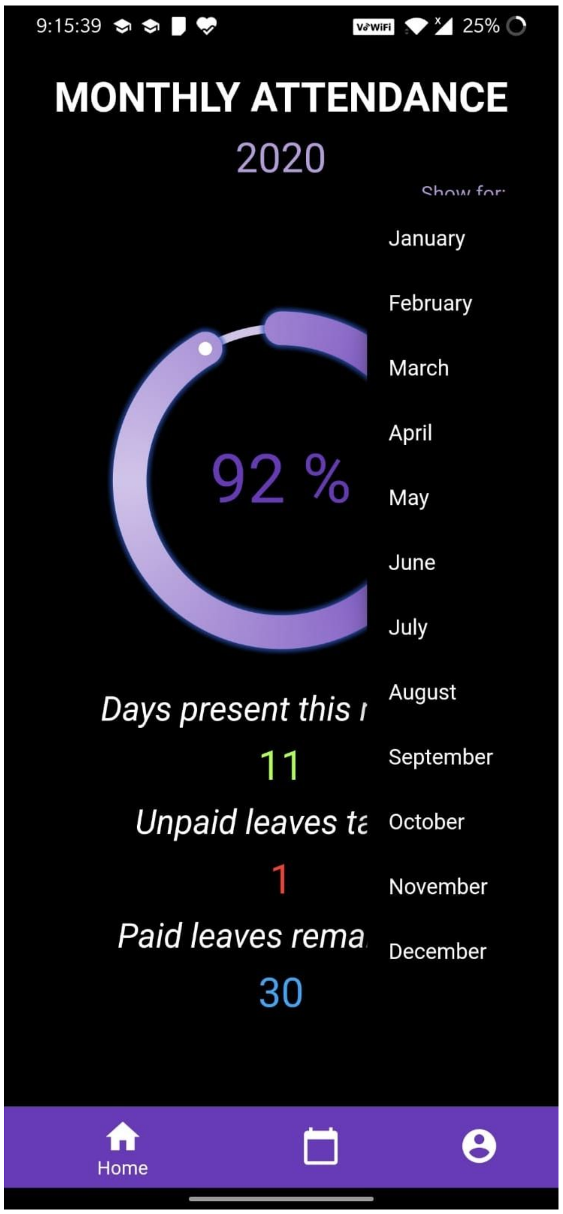
- Firebase Storage contains the folder for the images of the employees which are matched against the pictures taken every day by the office cameras.

The methodology flow begins with the pre-processing of a photo of the individual, then facial feature extraction from the photo, subjective selection, classification of the facial features to be recognized.
A camera detects the faces of individuals. These faces are then matched using AI to the available images, useful features extracted from the images, and classification done to identify the individual. Then their attendance is marked using the details that are retrieved from the organization's database.

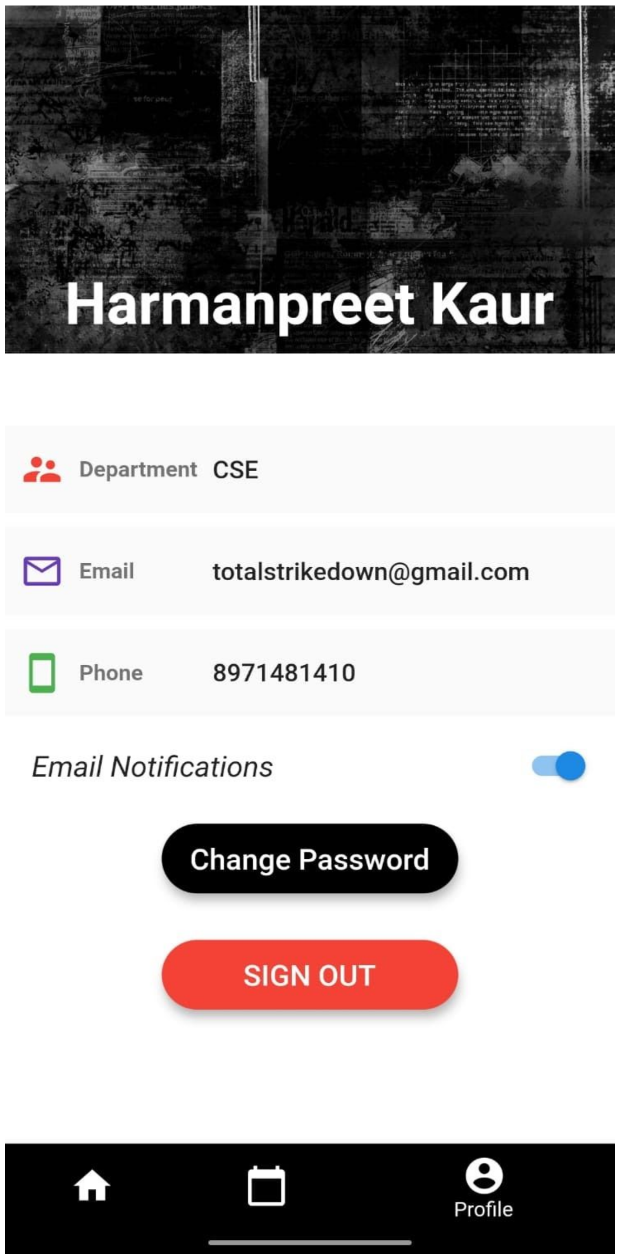
An email notification is sent to the person confirming that his/her attendance has been marked and the changes updated in the database.
An app that will display the details of the attendance is available, which will have a private log-in so that every individual can see his/her record.

This app has a home-page that displays the monthly attendance percentage, the number of days present, monthly unpaid leaves taken and annual paid leaves left. A drop-down list of all the months enables the user to navigate to and check the required month's details. There is a navigation bar at the bottom containing two more buttons for the calendar tab and the profile tab.





The **profile tab** contains the user's name, email id, department, and phone number. A switch button to enable email notifications is also included, so is a "Change Password" button. A sign-out button signs out the user. Unless the person signs out he/she will be kept logged in.

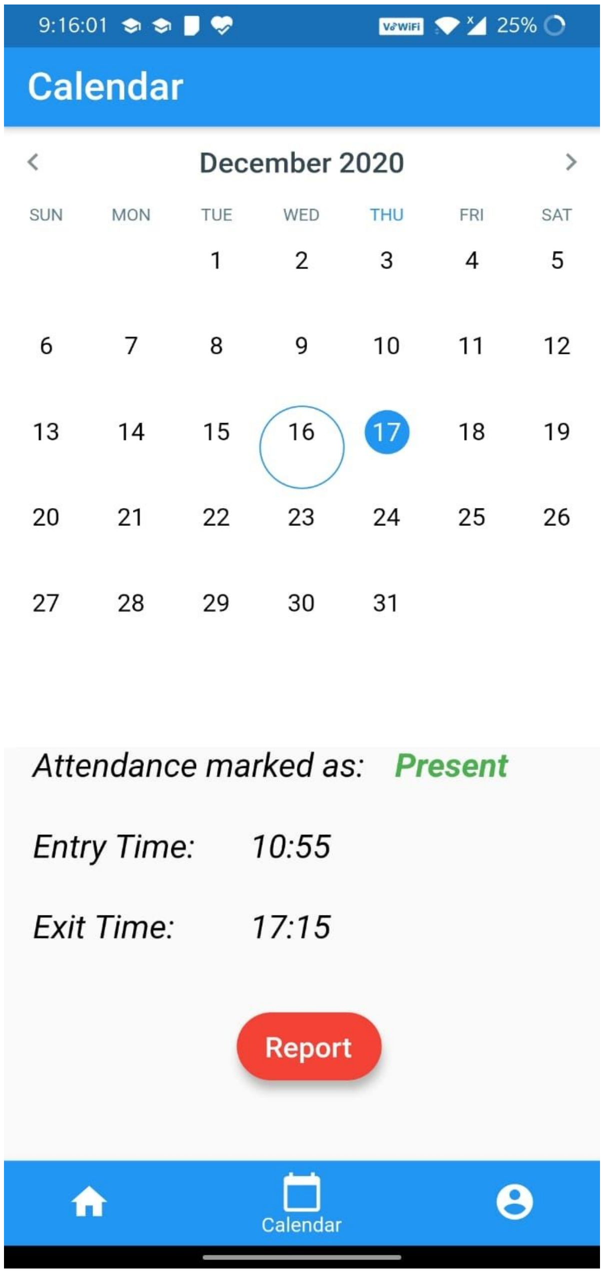


Harmanpreet Kaur

Department	CSE
Email	totalstrikedown@gmail.com
Phone	8971481410

Email Notifications

The calendar tab contains detailed information for each day of the year, displaying the attendance status for the day, and if the person were present, the entry and exit time.



SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Attendance marked as: ***Present***

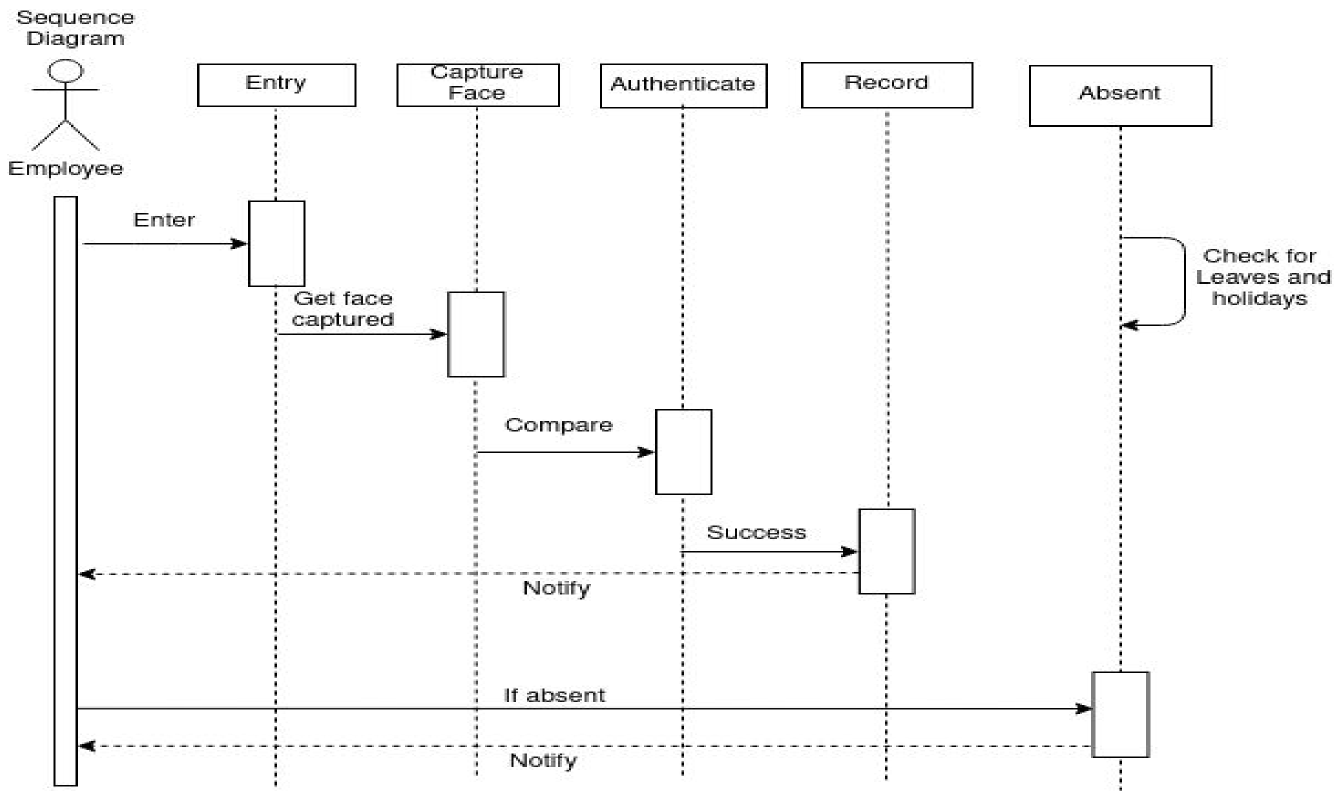
Entry Time: *10:55*

Exit Time: *17:15*

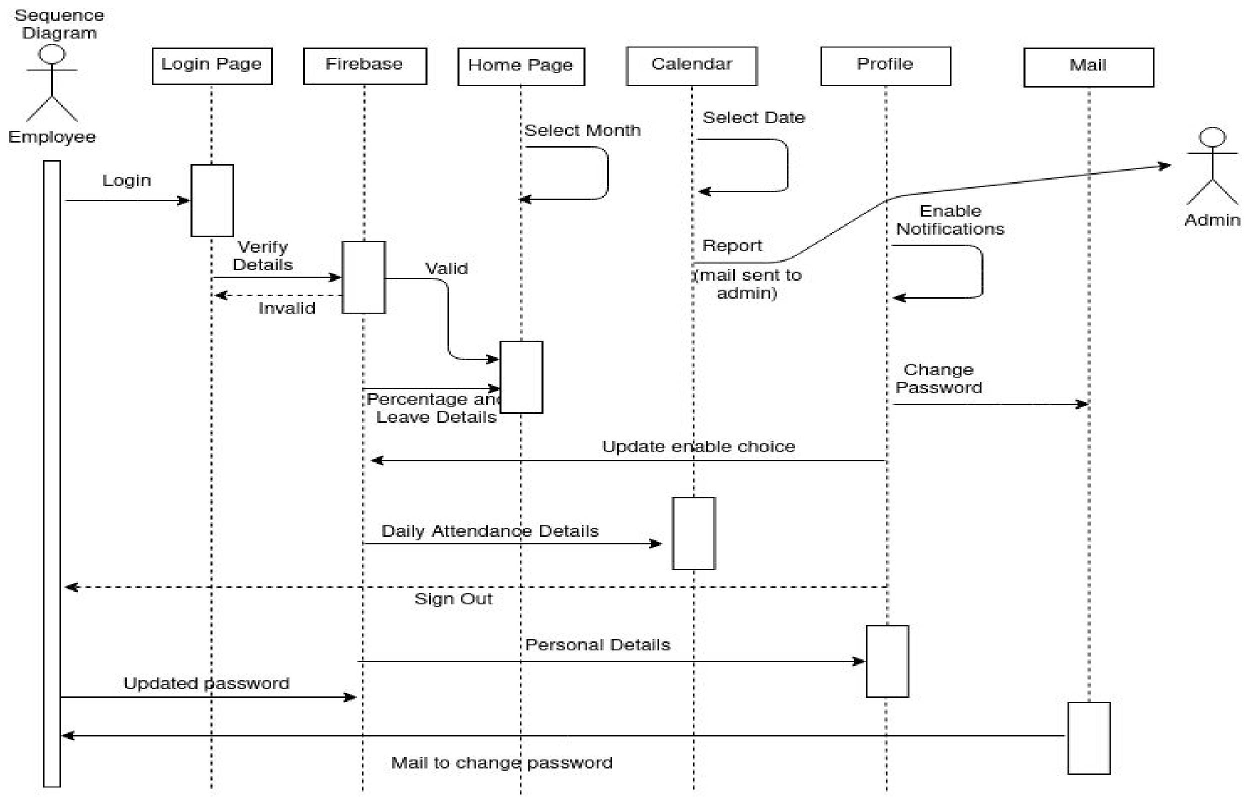
Incorporation of Software Engineering:

Software Engineering forms the basis of all projects as it entails how to go about developing it. The process model which we have used is the Iterated model or Incremental delivery model. We added requirements as we were developing the project and in each iteration, the design, coding and testing have been done.

1. Sequence diagrams
   a. Marking attendance



b. App



Deliverables

We have an efficient and no-touch automated attendance marking system based upon facial recognition with improved features like automated compilation of recorded attendance, elimination of proxy, etc. along with an android app that previews the entire record of the attendance data. Additionally, a downloadable excel sheet that contains this data for all the employees.

Conclusion

During the duration of the project, we accomplished our goal of developing a system for marking attendance and previewing the data on an android application. This attendance system will allow organisations to do away with traditional attendance systems. In order to achieve the best results we have also implemented the auto border detection for the document so that the application can automatically detect the borders and edges of the scanned image.

References

- https://pypi.org/project/face-recognition/
- https://firebase.google.com/docs
- https://pub.dev/packages/syncfusion_flutter_calendar