

Attendance Management System Using Facial Recognition

Team No - 28

Nimisha Srivastava

Gaurav Aswani

Arshad Mohammed

Ritabrata Podder

Sangram Keshari Dash

Pushpa Yadav

Ramakrishna

Project Abstract

Recording and managing attendance is an unavoidable overhead for any lecture. As much as attendance is necessary, it consumes a lot of productive lecture time, especially for larger classes. The crucial class-time can be saved with a student attendance system, which is an automated solution to not only mark attendance but also to find out meaningful patterns in order to monitor student performance around key metrics.

This project is an innovative solution to automate the process of recording attendance by facial recognition of students present in the class through one or multiple class pictures. We define a workflow around the generation of the attendance report as approved by the faculty and provide key metrics around the recorded attendance with automatic alerts sent to the stakeholders from time to time.

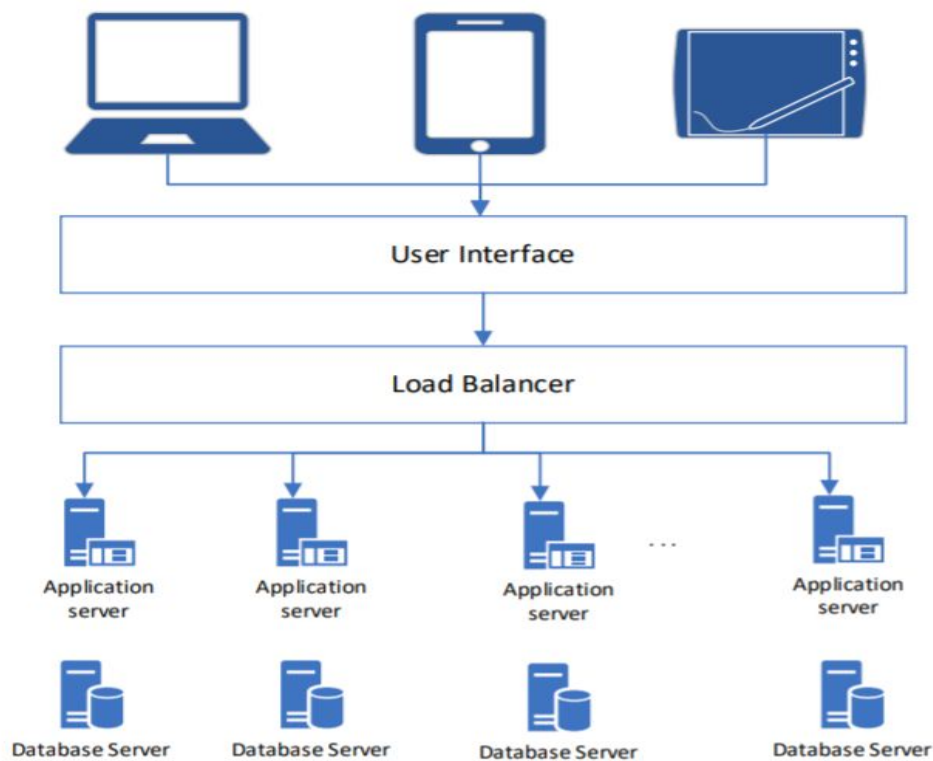
Goals

1. Ease the process of recording attendance by giving a simple, intuitive platform to upload images of the class to generate attendance for a day.
2. Give the system admin and the faculty the flexibility to address any discrepancies in the recorded attendance.
3. Prompt necessary action to admin and faculty and trigger notifications to all stakeholders.
4. Show key metrics around attendance to provide necessary indicators to the stakeholders
5. System developed should be scalable to any number of courses and a wide student and faculty base
6. System should be adaptable to different platforms and screen sizes.
7. System should capture the maximum data in order to provide any metric around the attendance upon request.
8. Enable fast search by partitioning the active data and data of the semesters completed.

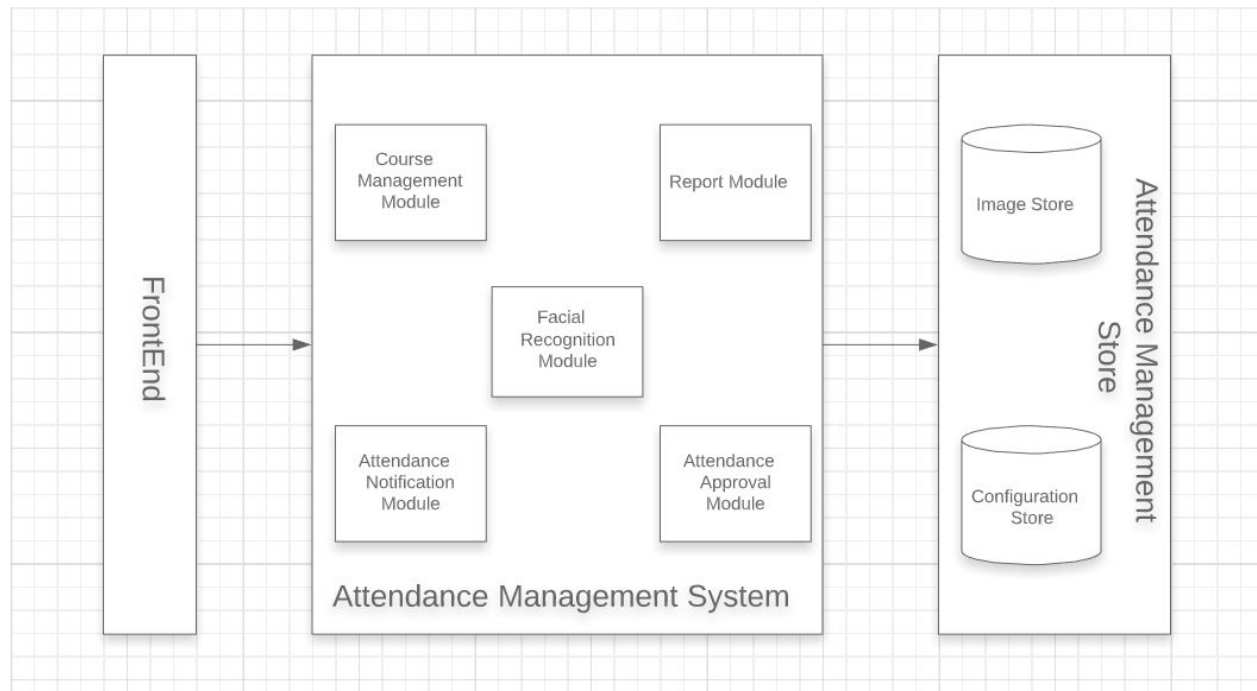
Assumptions

1. Student, Faculty and Course Master data will be pulled from the Institute systems.
2. The Face Detection API has been trained minimalistically as of now with little data. We assume that we will capture upto 10 pictures of different profiles of every student at the beginning of a semester(possibly as a part of the registration process, independent of this application) and train the algorithm separately. The application is made scalable to add this process of data capturing and training from within the application in the future.

High-Level System Design Diagram



Key Components of the Proposed Solution



This Attendance management system can be divided into 4 different workflows:

1. **Course Management Module** : This module is responsible for adding/modifying details of the courses available.
2. **Facial Recognition Module** : This is the core module that enables Facial recognition of a student's image in the class image(s) to record his/her attendance.
3. **Attendance Notification Module** : This module provides functionality of notifying faculty or students for various scenarios like notifying a student of a missed class.
4. **Attendance Approval Module** : This module will take care of the approval workflow for enhancing the attendance results by the facial recognition module.
5. **Report Module** : This will provide configurable reports on key metrics to the course faculty to analyze the performances of students.

Overview of the System Architecture

Deploy the facial recognition algorithm web app onto the amazon cloud and use REST services in AWS lambda to communicate between the client and DynamoDB via Amazon API gateway.

Our application leverages available AWS cloud architecture consisting of AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB, Amazon Cognito, and AWS Amplify Console applications.

Amplify Console hosts static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser via S3.

JavaScript executed in the browser sends and receives data from a public backend API built using Lambda and API Gateway. Amazon Cognito provides user management and authentication functions to secure the backend API. Finally, DynamoDB provides a persistence layer where data can be stored by the API's Lambda function.

We will also be using the Amazon Rekognition API to analyze the uploaded images and save picture metadata to the database.

