MANIT

NIT Bhopal

CA 302

Software Architecture

Assignment on

Functional and Non-functional Requirements

by

Vishal Srivastav

Scholar No: 202120038

Presented to

Prof. Priyanka Upadhaya

Introduction to Project

Smart Attendance system

Project Scope:

Our project (currently in development by Ms. Nidhi Sharma and myself) is designed to leverage the capabilities of ML algorithms that can identify and recognize faces. This feature is used by a camera module to recognize faces in pictures taken by it and mark the attendance of standing individuals. It will contain admin functionalities to add more students/employees, a user interface for basic interaction with an automated module that will handle day-to-day attendance with great accuracy.

Objective:

→ To develop a portable Smart Attendance System which is handy and self-powered.

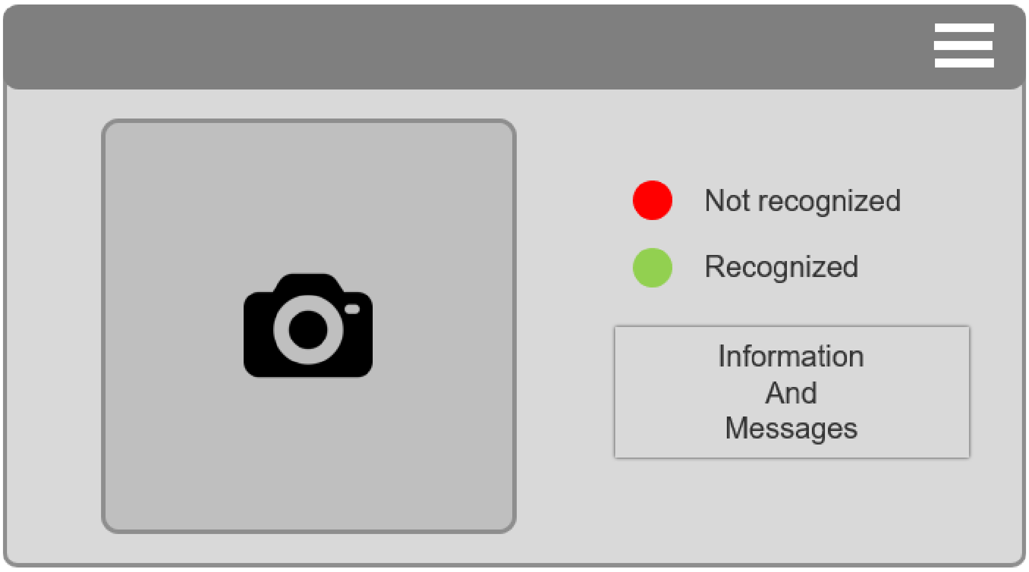
→ Able to recognize the face of an individual accurately

→ Provide a user-friendly admins panel

→ Allow admins to add new faces to the database

→ Live indication of successful recognition or failure

Software Design(expected):



Requirements

Functional

Capturing Image:

Function: capture images using the webcam of device
Priority: Top(Core functionality)
Requirements: we require WebRTC, Jpegcam, or webcam like modules/libraries in order to capture images using Django framework

Face Recognition:

Function: Recognize faces in pictures
Priority: Top(Core functionality)
Requirements: In order to recognize faces in pictures taken by camera module we will require complex ML algorithms. To achieve that we will use Python library OpenCV, Pandas and NumPy.

Success/Failure Response:

Function: Provide user with visual response of successful or failed recognition.
Priority: Top(Core functionality)
Requirements: the successful working of above two modules will be enough for generation of this response.

Adding User:

Function: Add a new user to the system(admin functionality)
Priority: Top(Core functionality)
Requirements: admin privileges is required. Images of new user is required to train the module.

Display Information:

Function: Display information of all the users(admin functionality)
Priority: High(Required for proper management)
Requirements: admin privileges is required. A relational database is required to retrieve the data from.

Keep track of attendance:

Function: manage the attendance based on the information generated by face recognition module
Priority: Top(Core functionality)
Requirements: admin privileges is required. Images of new user is required to train the module.

Mail reports:

Function: The attendance report generated will be mailed to concerned authority
Priority: Top(Core functionality)
Requirements: This will be achieved by automated module that will incorporate python libraries to send automated mails.

Non-functional

Security:

Function: administration access must require proper validation
Priority: Top(Core functionality)
Requirements: Django admi module

User access:

Function: user must be restricted to main system and shall not be able to access the internal component and database.
Priority: High(Security related module)
Requirements: Django is used to implement the security boundaries.

Reliability:

Function: The accuracy of this system must be very accurate

Maintainability:

Function: It shall provide functionality to add, remove and update user/user information.
Requirements: admin module will be required to implement entire CRUD operation

Performance Requirement:

Function: should work concurrently, should generated success/failure response within 5 sec.
Priority: High
Requirements: Django admi module

Software Interface:

- → Home screen should be user friendly
- → Admin module must be secure and provide desired functionalities
- → Display records and user information
- → Help option: it will help new users to understand and use the system