# 7. POC

## 7.1 Point of Contact (POC)

The system implements a specialized user type called POC which extends the base User model. POCs have additional fields for job role and contact information, along with a verification system supporting both email and mobile verification methods.

Key Features: 1. POC Registration 2. POC Authentication 3. Verification System 4. Temporary Password Generation 5. JWT Token Management

### 7.1.1 POC Management

#### Pre-Condition

* Authenticated PM required for POC registration
* Valid email and phone number required
* Verification method must be specified

#### Involved Users

| User Role | Activity Responsibility |
| --- | --- |
| PM | • Register new POCs |
|  | • Manage POC accounts |
| POC | • Login to system |
|  | • Change Password |

#### Fields and Validations

| Field | Description | Type | Validation Rules |
| --- | --- | --- | --- |
| username | Unique username | Textbox | • Required |
|  |  |  | • Unique |
| email | Email address | Email | • Required |
|  |  |  | • Valid email format |
| job\_role | POC’s job role | Textbox | • Optional |
|  |  |  | • Max length 100 |
| phone\_number | Contact number | Textbox | • Required |
|  |  |  | • Max length 15 |
| verification\_method | Preferred verification method | Choice | • Required |
|  |  |  | • Options: email/mobile |
| verified | Verification status | Boolean | • Default: False |

#### API Endpoints

1. **Register POC**

POST /api/poc/register/  
Authorization: Required  
Permission: IsPM  
**Body:**  
 {  
 "email": "string",  
 "username": "string",  
 "job\_role": "string",  
 "phone\_number": "string",  
 "verification\_method": "email|mobile"  
}

**Response:**

{

"message": "POC registered successfully! Password sent to registered email."

}

1. **POC Login**

POST /api/poc/login/  
Authorization: Required  
Permission: IsPM  
**Body:**   
{  
 "email": "string",  
 "password": "OTP"  
}

**Response:**

{

    "refresh": "Refresh\_token",

    "access": "Access\_token"

}

1. **Change Password**

POST authentications/change-password/  
Authorization: Required  
**Body:**

{

  "old\_password": "OTP",

  "new\_password": "new\_pwd"

}

**Response:**

{

    "status": 200,

    "message": "Password updated successfully."

}

#### Activity Diagram

#### Error Messages

| Sl No | Message Text | Scenario |
| --- | --- | --- |
| 1 | Invalid email or password | During login with incorrect credentials |
| 2 | Email is required | When email is missing in registration |
| 3 | Phone number is required | When phone number is missing |
| 4 | Invalid verification method | When invalid method is selected |
| 5 | Email sending failed | When verification email fails |

# Face Authentication and Attendance System with Geo-Location - Documentation

## Overview

This is a Flask-based application for face authentication, attendance tracking, and location tracking. It integrates face recognition using MediaPipe, geolocation tracking, and reporting features.

## Installation & Setup

### Prerequisites

Python 3.x  
Flask  
Flask-Login  
Flask-SQLAlchemy  
OpenCV  
MediaPipe  
NumPy  
ReportLab  
Pillow  
Folium  
pytz  
smtplib

### Setup

1. Install dependencies:  
```sh  
pip install flask flask-login flask-sqlalchemy opencv-python mediapipe numpy reportlab pillow folium pytz smtplib  
```  
2. Run the application:  
```sh  
python app.py  
```

## Database Models

### User Model (User)

Contains user information such as name, email, password, face embedding, and contact details.

### Attendance Model (Attendance)

Stores user attendance records with timestamps and geolocation data.

### Location Track Model (LocationTrack)

Stores user location tracking data with timestamps and coordinates.

## API Endpoints

### User Authentication

Endpoints for user login, registration, and logout.

### Attendance Tracking

Endpoints for marking attendance with face authentication and geolocation.

### Location Tracking

Endpoints for tracking user location and viewing it on a map.

### Attendance Reports

Endpoint for generating attendance reports in PDF format.

## Utility Functions

Includes face recognition, email notifications, and PDF generation functions.

## Conclusion

This system ensures secure face-based authentication with attendance and geolocation tracking. It provides automated report generation and email delivery for streamlined attendance management.