FRAS – Test Plan

1. Project Information

Project Name: FRAS
Team Name: Group 3
Prepared by: Brendon Stepanek, Maximiliano Hernandez, Joshua Odegai, Neel Patel, Zain Jamal
Date: 10/4/2025

2. Introduction

This document outlines the test plan for our facial recognition attendance system. Its purpose is to verify that all system functions and components meet the functional and non-functional requirements that are outlined in our SRS document.

3. Test Objectives

- Test and verify that all features work as intended.
- Identify and document any bugs and ensure that they are fixed.
- Test usability and performance of the system.
- Test the system's stability and accuracy in real-world situations.

4. Testable Items

4.1 In-Scope

- Login/Logout & User Authentication
- Instructor Functions
    - View Classes
    - View Class Attendance
    - Modify Attendance Records
    - Export Attendance Records

- Student Functions
    - Facial Recognition & Attendance Verification
    - View Classes
    - View Individual Class Attendance
- Admin Functions
    - Add Class
    - Add User
    - Manage User
    - Manage Class
- AI Notifications
    - Attendance/Class Reminders
    - Notification to instructor if student has > 5 absences
- IP Geofencing/Network Verification
- Dark/Light Mode Toggle

4.2 Out-of-Scope

N/A – All implemented features are necessary to our project and within the scope of our project, therefore should be tested.

4.3 Functional Scope

- User Authentication
    - Login/Logout
- Facial recognition & Attendance Recording
- Manual Attendance Record Modification
- AI Notifications
    - Attendance/Class Reminders

    - Alerts for Instructors
- IP Geofencing & Network Verification
- User & Class Management for Admins
- Attendance Record Exports
- Theme Switching
    - Dark and Light mode

4.4 Actual Components to be Tested

Functional Area		Description
Frontend UI	**Login Page**	**Email and password input, input errors, redirect on successful login**
	Student Dashboard	**Class list, class attendance view, notifications inbox**
	Instructor Dashboard	**Class List, class attendance view, manual record modification, notifications inbox**
	Face Scanner	**Webcam capture, feedback/status from scans**
	Reports Page	**Data filter, CSV export button**
	Notifications UI	**In app banners for reminders/updates**
	Theme Toggle	**Dark/Light mode state**
Backend API	**Facial Recognition Endpoint**	**Processes and compares the stored image with the temp image captured of the student.**
	Attendance Confirmation Endpoint	**Implements the 30-minute verification window for student attendance**
	Attendance Data Get/Push	**Gets/Updates student attendance records**
	Class & Class Roster Endpoints	**Get list of classes, fetches class details, and all enrolled students**
	Reports/Exportation Endpoint	**Generates an attendance report for specific class in CSV form file**
	Geofence Verification Endpoint	**Validates the user's network and checks IP boundary**
Database Interactions	**Users**	**userID, role, classes, email, first and last name**
	Classes	**course related metadata, schedule, userID**
	AttendanceRecords	**classID, userID, timestamps, status, geofence status**
	Notifications	**Sent reminders and alert history**
	ErrorLogs	**Log all errors and important system events**
Authentication	**Firebase**	**Login/Logout**

	Authentication	
	Role-based Access Control	**Routing and API for roles**
AI Notifications	**Class Reminder**	**Reminder of class start 5 minutes before scheduled start time**
	Absence Threshold Alerts	**Notify instructor if a student has > 5 recorded absences**
IP Geofencing	**Network Verification**	**IP, allowed IP range, logging of invalid Ips**
Error Logging	**Error Logs**	**API errors, recognition failures**

4.5 Non-Functional Scope

List all non-functional areas and descriptions that will be in scope of testing.
Examples:
- Performance: system response under normal load
- Usability: clarity of error messages and navigation
- Compatibility: works on Chrome and Firefox

Table 1: Non Functional

Non Functional Requirement	Description
NFR-001: System Performance	**Attendance recognition and verification process should be completed within 5 seconds under acceptable conditions.**
NFR-002: Usability	**The system's UI should be simple, clean, responsive, and intuitive for all users.**
NFR-003: Data Security	**All sensitive user data should be encrypted.**
NFR-004: System Reliability	**Our system should maintain at least 99% uptime.**

5. Test Cases

Test Case 1: Valid Username and Password

Description: Verify login with valid user credentials.

Test Steps:

- Navigate to login page.
- Enter valid email and password.
- Click "Login."

Expected Outcome: Redirects to correct dashboard depending on user role.

Test Case 2: Invalid Login Credentials

Description: Reject invalid login attempts.

Test Steps:

- Navigate to login page.
- Enter incorrect/invalid login credentials.
- Click “Login.”

Expected Outcome: Displays “Invalid credentials” and user remains on login page.

Test Case 3: Logout

Description: Ensure users can log out successfully.

Test Steps:

- Log in.
- Select “Sign out” on dashboard page.

Expected Outcome: Returns to login screen.

Test Case 4: Facial Recognition – Successful Match

Description: Recognized student is given a confirmation message of successful scan.

Test Steps:

- Open scan from student dashboard.
- Select a class.
- Center face in camera and start scan.

Expected Outcome: Student is recognized and confirmation/status pending message is given to user.

Test Case 5: Facial Recognition – Unrecognized Face

Description: Handle unknown faces properly.

Test Steps:

- Attempt scan process with an unrecognized face.

Expected Outcome: Displays “Face not recognized”, event logged, no record is created.

Test Case 6: Manual Attendance Modification (Instructor)

Description: Instructor can modify attendance manually.

Test Steps:

- Log in as instructor.
- Open class attendance page.
- Select a student and change their status.

Expected Outcome: Record updates and override is logged.

Test Case 7: View Classes (Instructor)

Description: Instructor views their assigned classes.

Test Steps:

- Log in as instructor.
- Navigate to “My Classes” page from dashboard.

Expected Outcome: Displays correct class list.

Test Case 8: View Class Attendance (Instructor)

Description: Instructor views attendance table.

Test Steps:

- Log in as instructor.
- Navigate to “My Classes” page from dashboard.
- Select a class.

Expected Outcome: Accurate attendance list and chart shown.

Test Case 9: Student – View Own Attendance

Description: Student can view their own records.

Test Steps:

- Log in as student.
- Open “My Classes”
- Select a class.

Expected Outcome: Displays only that student’s attendance records.

Test Case 10: Reports – Generate by Date Range

Description: Create accurate attendance reports.

Test Steps:

- Log in as Instructor.
- Go to “My Classes” page.
- Select a class.
- Choose a date range and click “Export.”

Expected Outcome: Shows correct totals matching stored records.

Test Case 11: Geofencing – Outside Allowed Network

Description: Block attendance submission.

Test Steps:

- Connect from non-approved IP range.
- Attempt to submit attendance.

Expected Outcome: Submission is blocked and violation logged.

Test Case 12: Geofencing – Inside Allowed Network

Description: Allow attendance from approved IP.

Test Steps:

- Connect from approved IP
- Submit attendance.

Expected Outcome: Successfully records attendance; logs geofence status as inside.

Test Case 13: AI Notification – Pre-Class Reminder

Description: Send notification 5 minutes before class.

Test Steps:

- Ensure a class is scheduled today.
- Receive notifications at class start time minus 5 minutes.

Expected Outcome: Reminder delivered and logged.

Test Case 14: AI Notification – Absence Threshold Alert

Description: Notify instructor when a student has 5 absences.

Test Steps:

- Use student with 5 absences.
- Trigger notification process.

Expected Outcome: Instructor alert sent, and event logged.

Test Case 15: 30-Minute Pending Rule

Description: Confirm finalization timing.

Test Steps:

- Student scans successfully.
- Wait 30 minutes within geofence.

Expected Outcome: Marked Present, leaving early before 30 min sets to Absent.

Test Case 16: Performance – Multiple, Concurrent Scans

Description: Maintain stability under high load.

Test Steps:

- Simulate 10-25 simultaneous scans.

Expected Outcome: Average recognition time under 5 seconds, no system crash/errors, or data loss.

Test Case 17: Error Handling – Network Drop During Scan Process

Description: Verify handling after network failure.

Test Steps:

- Start a scan.
- Disconnect from Wi-Fi network mid-scan and reconnect.

Expected Outcome: Error logged, and user can retry scan.

Test Case 18: Unauthorized Access Blocked

Description: Restrict role-based access.

Test Steps:

- Log in as a student.
- Attempt to access Instructor/Admin pages.

Expected Outcome: Access denied, proper error shown, no unauthorized data access.

Test Case 19: Facial Recognition – Multiple Faces in Frame

Description: Prevent scanning when multiple faces are visible.

Test Steps:

- Two students stand in camera view.
- Start scan.

Expected Outcome: System displays error message and blocks scan.

Test Case 20: Camera Permission Denied

Description: Handle lack of camera access properly.

Test Steps:

- Deny camera permission in browser.
- Open scan page.

Expected Outcome: Scanning disabled until camera permission is granted.

Test Case 21: Duplicate Scan Suppression

Description: Prevent duplicate creation of attendance records.

Test Steps:

- Complete a scan.
- Attempt multiple scans.

Expected Outcome: Only one attendance record is recorded, and others are blocked.

Test Case 22: Finalize Window (30 Minutes)

Description: Check system behavior at finalize threshold.

Test Steps:

- Student scans and leaves at 29 minutes.
- Another student stays for 31 minutes.

Expected Outcome: First student marked “Absent” and second student marked “Present.”

Test Case 23: Geofencing with IP change

Description: Manage network changes.

Test Steps:

- Connect from approved IP and start attendance recording process.
- Switch to hotspot or VPN and return.

Expected Outcome: Handles and logs network change and maintains accurate tracking.

Test Case 24: Large Class Report Export

Description: Verify report accuracy with large number of students.

Test Steps:

- Use class with 75+ students.
- Generate and export report.

Expected Outcome: Data is accurate across all students and exports complete.

Test Case 25: API Slow or Failure with Retry

Description: Ensure proper handling of slow API or outages.

Test Steps:

- Simulate slow or failed API response.
- Attempt scan.

Expected Outcome: Displays retry or error message and logs issue.

Test Case 26: Add New User (Admin)

Description: Verify that admin role can create a user.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Users” page from dashboard.
- Click “Add User.”
- Enter user’s information and assign role.
- Click “Save.”

Expected Outcome: The new user is created and appears in the list of users with correct data and roles.

Test Case 27: Edit User (Admin)

Description: Verify that an admin can edit user information/roles.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Users” page from dashboard.
- Select a user.
- Update user information.
- Click “Save.”

Expected Outcome: The system saves the new user information and is reflected in the list of users.

Test Case 28: Delete User (Admin)

Description: Verify that an admin can delete a user.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Users” page from dashboard.
- Select a user.

- Click “Delete.”

Expected Outcome: The user is deleted from the system and is no longer in the list of users.

Test Case 29: Add New Class (Admin)

Description: Verify that admin role can create a class.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Classes” page from dashboard.
- Click “Add Class.”
- Enter class information.
- Click “Save.”

Expected Outcome: The new class is created and appears in the list of classes with correct data.

Test Case 30: Edit Class (Admin)

Description: Verify that an admin can edit user information/roles.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Classes” page from dashboard.
- Select a class.
- Update class information.
- Click “Save.”

Expected Outcome: The system saves the new class information and is reflected in the list of classes.

Test Case 31: Delete Class (Admin)

Description: Verify that an admin can delete a user.

Test Steps:

- Log in as an administrator.
- Navigate to “Manage Classes” page from dashboard.
- Select a class.
- Click “Delete.”

Expected Outcome: The class is deleted from the system and is no longer in the list of classes.

6. Test Approach

We intend to do both manual and automated testing where possible and log every result.

- Unit Testing
    - We'll be testing each module individually.
- Integration Testing
    - To ensure proper data flow, we'll be sure to test system integration across our frontend, backend, and database.
- Functional Testing
    - Each individual feature will be tested against the requirements stated in our SRS document.
- Performance Testing
    - We'll test the system's efficiency and reliability under high user loads.

7. Test Schedule

Activity	Function to be Executed	planned Date
Test Plan Approval		**10/5/2025**
Test Execution Start	• **Login** • **Attendance Verification** • **AI Notifications** • **Geofencing**	**10/6/2025**
Test Execution Ends	• **All Functional & Non-Functional Areas**	**10/8/2025**
Final Report Submission	• **Test Summary** • **Documentation**	**10/12/2025**

8. Entry and Exit Criteria

8.1 Entry Criteria

- All features are deployed to the testing environment.
- Test cases are prepared.

8.2 Exit Criteria

- All test cases are completed.
- All issues/bugs are resolved.
- Test summary is complete.
- Team and stakeholder review and sign off.

9. Pass/Fail Criteria

- We'll deem a test case as "pass" when the feature is confirmed working as expected and meets all acceptance criteria.
- We'll deem a test case as "fail" is the feature does not function as intended.

10. Test Environment

Testing will begin on our production environment to create a baseline. As issues/bugs are found, fixes will be implemented to a local/development build and tested again. Once an issue is resolved, fixes will be pushed to the production environment.

- Web App
    - Chrome & Safari
    - Firebase Firestore
    - Localhost Server

11. Roles and Responsibilities

Team Member	Role	Responsibilities

Brendon Stepanek	Team Lead / Developer / QA	• Assisting with backend and frontend testing • Resolve reported bugs • Ensure all issues are resolved before deployment
Zain Jamal	QA / Documentation	• Records and tracks all test findings/results • Maintains documentation of all test cases and their outcomes • Will assist will UAT prep and reporting
Maximiliano Hernandez	Frontend Tester / Documentation	• Focusing on frontend testing such as UI functionality, consistency, and user experience • Validating that all user pages and functionality work
Neel Patel	Developer / Backend Tester	• Performance testing for frontend, backend, and database • Run load and stress tests to verify system reliability • Debugging, fixing system errors or performance issues
Joshua Odegai	Lead Tester / Frontend & Backend QA	• Leading the testing phase • Overseeing test execution and documentation • Coordinating with team members to ensure all test cases are completed • Testing backend API endpoints • Data accuracy • Reports issues/bugs to development