Group 3

FACIAL RECOGNITION ATTENDANCE SYSTEM

TEST PLAN

Date: 3/23/25

Table of Contents

INTRODUCTION **3**

1.1 OBJECTIVES 3
1.2 TEAM MEMBERS 3

2 SCOPE **3**

3 ASSUMPTIONS / RISKS **5**

3.1 ASSUMPTIONS 5
3.2 RISKS 5

4 TEST APPROACH **6**

4.1 TEST AUTOMATION 7

5 TEST ENVIRONMENT **11**

6 MILESTONES / DELIVERABLES ..... ERROR! BOOKMARK NOT DEFINED.

6.1 TEST SCHEDULE ..... **ERROR! BOOKMARK NOT DEFINED.**
6.2 DELIVERABLES ..... **ERROR! BOOKMARK NOT DEFINED.**

Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

1.1 Objectives

The ultimate goal of our Facial Recognition Attendance System is to automate the attendance-taking process and remove the inconvenience and hassle that it creates.

Our usability test plan will focus on the following:

- Login and User Authentication
  - Users need to be properly directed to their respective pages based on their role. (Student, Teacher, Administrator)
- UI Navigation
  - We need to test the overall user experience and usability of each of our webpages.
  - We should confirm that all buttons function as they should and all links lead to the expected pages.
  - Responsiveness and error handling also need to be thoroughly evaluated.
- Facial Recognition Process
  - We should assess the accuracy and efficiency of the facial recognition system, ensuring that attendance records are updated properly.

1.2 Team Members

Resource Name	**Role *(examples are given below)***
Brendon Stepanek	Developer, Tester, Project Manager
Josh Odegai	Developer, Tester
Zain Jamal	Developer, Tester
Maximiliano Hernandez	Developer, Tester
Neel Patel	Developer, Tester

2 Scope

The initial sprint will include ‘must have’ requirements. These and any other requirements that get included must all be tested.

The following sections indicate what is tested during each sprint. The scope of testing is determined at the beginning of the current sprint.

At the end of Sprint 1, a user must be able to:

- Log in using an email and password and be taken to the relevant dashboard depending on the user’s role.
- Use the Teacher’s dashboard to understand the process of viewing attendance for classes and sending messages.
- Properly navigate teacher pages when the user clicks on different UI elements like Dashboard, My Classes, and Messages.

At the end of Sprint 2, a user must be able to:

- Use the facial recognition system to record attendance records accurately.
- Be able to view attendance records.

- Have role-based permissions to allow each role access to their respective features.
- See basic attendance metrics generated by the system.

At the end of Sprint 3, a user must be able to:

- Complete the entire end to end workflow for the attendance tracking process, from login all the way to attendance confirmation.
- Create and manage classes, assign user roles, and generate reports.
- Use a final, polished UI based on any user feedback from earlier testing stages.

Assumptions / Risks

2.1 Assumptions

- Users will access the system on a Windows PC.
- The facial recognition models DeepFace and OpenCV perform as expected.
- The system will be used in an environment with adequate lighting.

2.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Scope Creep – as testers become more familiar with the tool, they will want more functionality	High	Delays in implementation date	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out.
2	Accuracy of facial recognition is affected by poor lighting conditions	High	Low accuracy in attendance detection	Layout guidelines/requirements for optimal lighting conditions
3	System performance suffers under high load	High	Slow system response time when multiple users are using the system simultaneously	Conduct load testing and optimize backend as best as possible
4	Vulnerabilities in Authentication and Data Storage	High	Unauthorized access/breaches	Encrypt data and conduct proper penetration testing
5	Facial Recognition False Positive/Negatives	High	System does not properly mark attendance	Improve model training, allow manual attendance modification
6	Hardware Dependency	Medium	System may fail to work on low quality cameras	Set minimum camera resolution requirements
7	User Resistance	Medium	Users may find system difficult to use	Create user training documentation/guides and gather feedback for improvements

3 Test Approach

The project is using an agile approach, with 3-week sprints. *Mention how you will conduct testing during the sprint in terms of the techniques you plan to do and when. Add a new subsection for each sprint.*

Sprint 1

- Testing in this sprint mostly focused on login authentication, navigating the UI, testing basic webpage elements and dashboard access.
    - Access the dashboard based on their user role and validate that users are logged in correctly.
    - Test error handling for incorrect login attempts.
    - Test that all buttons, scroll bars, and search bars function as they should.

Sprint 2

- Testing in this sprint will focus on the facial recognition process, role-based permissions for modifying records, view attendance, and attendance analytics.
    - We need to ensure that attendance records are updated and stored accurately after the facial recognition process has finished.
    - We need to make sure that changes to attendance records and updated and stored correctly.
    - We also need to validate that attendance records and analytics are displayed properly in the pie chart of each class and student page.

Sprint 3

- Testing in this sprint will focus on the end to end functionality of our system.
    - We're going to conduct complete workflow testing, starting from the login page to the attendance recording confirmation.
    - We're also going to test to make sure administrator actions are reflected in the system such as managing users and classes. We'll attempt to create classes and users, edit those users and classes, and assign roles to users as well as classes to users.
    - We can also perform security testing to confirm that data is indeed encrypted.

3.1 Test Automation

We can implement automated testing in a few parts of our project. We can use PyTest to create tests for API authentication and attendance updating. If we feel the need to automate testing for the UI, we can use Cypress or Selenium. These can be used to automate login flows and UI navigation. For performance, we can use Locust or JMeter for looking at system response time when under high load. JMeter may be the better option because of the built-in dashboard that it has.

3.2 Test Cases (Black Box)

3.2.1 Facial Recognition Attendance Tracking

Test Case ID	Description	Requirements Trace	Directions	Expected Output
TC-001	Facial scan for attendance	Facial Recognition	1. Student scans their face	Attendance is recorded and updated successfully
TC-002	Attendance record modification	Attendance record modification for teachers	1. Teacher selects student on class page. 2. Teacher chooses the date 3. Teacher chooses the new attendance status	Status is updated and reflected in attendance records

3.2.2 Webpage UI for Users

Test Case ID	Description	Requirements Trace	Directions	Expected Output
TC-101	Students can login and navigate the pages they have access to	Role-based access and UI Navigation	1. Student logs in 2. Student navigates through the dashboard 3. Student clicks on “My Classes” and navigates through the page 4. Student clicks on their classes and can view their attendance records for each class 5. Student clicks on “Messages” and is able to view messages between teachers. 6. Student clicks on a message and is able to respond to a teacher in one of their conversations	Student is able to navigate allowed sections, view attendance record and use the messaging system without errors
TC-102	Teachers can login and navigate the pages they have access to	Role-based access and UI Navigation	1. Teacher logs in 2. Teacher navigates through the dashboard 3. Teacher clicks on “My Classes” and navigates through the page 4. Teacher clicks on a class and is able to view a list of students 5. Teacher is able to click on a student and is able to view and edit their attendance record. 6. Teacher clicks on “Messages” and navigates through the page 7. Teacher clicks on a message and responds to a conversation with a student	Teacher is able to navigate allowed sections, view attendance records, edit student records, and use the messaging system without errors
TC-103	Administrators can login and navigate the pages they have access to	Role-based access and UI Navigation	1. Administrator logs in 2. Administrator navigates through the dashboard 3. Administrator clicks on “Manager Users” and navigates through the page 4. Administrator is able to add a new user and edit a user 5. Administrator clicks on “Manage Classes” and navigates through the page 6. Administrator is able to add and edit classes	Administrator is able to navigate and manage users and classes

3.2.3 Attendance Reports

Test Case ID	Description	Requirements Trace	Directions	Expected Output
TC-201	Generating Attendance Reports	Attendance record generation for teachers	1. Teacher selects class they want to generate report for	System creates attendance report
TC-202	Exporting Attendance Reports	Attendance record exporting for teachers	1. Teacher selects class they want to generate report for 2. Teacher clicks “Export Attendance” button on class page	Report is generated successfully and is available for download in PDF or CSV format.
TC-203	Viewing individual student attendance records	Attendance reporting	1. Teacher selects class for student they want report for 2. Teacher selects the student	Student's full attendance history is displayed

3.2.4 Messaging

Test Case ID	Description	Requirements Trace	Directions	Expected Output
TC-301	User can view conversations and send messages	Messaging	1. User selects “Messages” on nav bar 2. User clicks on the conversation they would like to view 3. User can read conversation and choose to send a message	User can receive messages in their inbox User can send a message

3.3 Test Cases (White Box) Sprint 1

3.3.1 Webpage UI for Users

Test Case ID	Description	Directions/Goals	Expected Output
TC-1001	Validate hardcoded login system	Enter hardcoded credentials in login form	User is redirected to the correct dashboard
TC-1002	Verify button functionality	Click on all navigation buttons	Each button redirects to the expected page or executes desires feature
TC-1003	Ensure elements display correctly	Load each page	All elements appear correctly
TC-1004	Check role-based navigation	Login as student, teacher, and admin .	Each role can only access its designated pages
TC-1005	Validate messaging feature	Navigate to messages page	Messages appear correctly

4 Test Environment

Our testing environment consists of a few hardware devices, a few software requirements, and some network requirements. Our test environment requirements are:

Hardware:
- A PC or Laptop device
- An external webcam or camera

Software:
- Windows 10/11
- A web browser
    - Chrome
    - Mozilla Firefox

- Testing Tools
    - Cypress/Selenium
    - Locust/JMeter
    - PyTest
    - Postman

- Development Tools
    - VSCode
    - Firebase

Network:
- Stable Wi-Fi connection

5 Test Schedule

Task Name	Start	Finish	Effort	Comments
Test Planning	1/27/25	3/22/25	81 days	Review requirements documents and define scope.
Review Requirements documents	1/27/25	2/9/25	13 days	Ensure test scenarios align with project requirements
Create initial test estimates	2/14/25	2/15/25	1 day	Estimate resources and effort required for testing
Learn new test resources	3/21/25	3/22/25	½ day	Research automated testing tools
First deploy to QA test environment	3/14/25	3/15/25	½ day	Started webserver to view prototype frontend
Functional testing – Sprint 1	3/15/25	3/27/25	12 days	Test hardcoded logins, ensure correct redirection. Validate webpage navigation, button functionality, and role-based UI access.

Iteration 2 deploy to QA test environment	3/16/25	3/17/25	2 days	Create a QA server and deploy the latest build, confirm connectivity and functionality of core functions.
Functional testing – Sprint 2	3/16/25	4/19/25	35 days	Test attendance scan, role-based page access and log any defects/bugs and retest.
System testing	4/12/25	4/16/25	5 days	Test end to end workflow from login to scan to record to reporting. Ensure data is being updated instantaneously as well.
Regression testing	4/14/25	4/16/25	3 days	Conduct initial sprint 1 testing again to ensure that any new code has no broken previously working features or pages.
Usability Testing	3/16/25	4/16/25	32 days	Collect UX feedback from other students/users/stakeholders. Consider and implement any reasonable feedback.
Resolution of final defects and final build testing	4/12/25	4/17/25	6 days	Find any remaining bugs in the prototype and deploy fixes.
Deploy to Staging environment	4/19/25	4/19/25	1 day	After performance testing and stakeholder approval, we'll do one final go live.
Performance testing	TBA	--	--	--
Release to Production	TBA	--	--	--