

Project Information

Project Name: Facial Recognition Attendance System (FRAS)

Client/Sponsor: Professor Diana Rabah

Team Members:

- Brendon Stepanek
- Zain Jamal
- Joshua Odegai
- Maximiliano Hernandez
- Neel Patel

Purpose & Goals

The purpose of our maintenance plan is to ensure that our FRAS system remains completely functional, reliable, and secure.

Our goals include:

1. Maintain 99.99% uptime and good performance for real-time attendance tracking.
2. Ensure that our stored data stays secure and our system meets all privacy compliance.
3. Conduct regular updates to all dependencies, frameworks, and APIs.
4. Be sure to monitor and fix any bugs or system failures as quickly as possible.
5. Provide system administrators with clear maintenance responsibilities and guidelines.

Scope of Maintenance

The following parts of our system will be maintained:

- Software
 - Flask backend
 - React frontend
 - Firebase Firestore
 - DeepFace and OpenCV libraries
- Data
 - Firestore databases
 - Attendance logs
 - User data
 - Any configuration files
- Documentation
 - System manuals
 - User training guides/manuals
- Security
 - Firebase authentication
 - Data/System backups
 - User access control

Roles & Responsibilities

The following roles are all hypotheticals if the system was to be implemented in a real-world scenario and have a complete team to maintain it.

Role	Responsibilities
System Administrator	<ul style="list-style-type: none">• Schedule and perform system backups• Monitor system uptime• Manage Firebase Database
Developers	<ul style="list-style-type: none">• Apply updates• Update all system dependencies• Fix bugs/issues• Maintain code repositories
Project Manager	<ul style="list-style-type: none">• Oversee maintenance the cycle• Coordinate with sponsor/client• Approve any new major changes or additions
Support	<ul style="list-style-type: none">• Respond to any user issues/reports• Document the incidents• Escalate any unresolved reports

Maintenance Schedule

Task	Frequency
System Backups	Daily
Dependencies and Security Updates	Monthly
Performance Testing	Quarterly
Facial Recognition Model Testing/Optimizations	Quarterly
User Account and Access Review	Quarterly
Review of Error Logs	Bi-weekly
Complete System Review/Audit	Semi-annual
Documentation Review	Semi-annual

Required Resources

- Software
 - Flask
 - React
 - Firebase
 - PyTest
 - GitHub
 - VS Code
 - Any internet browser
 - Chrome/Firefox/Safari

- Hardware
 - Local development computers
 - Test servers
 - Camera for testing purposes
- Budget
 - Currently, we're able to run our databases on a free tier of Firebase and our backend is running on a cheap, \$7 tier of Render. Though, if the system was to be implemented at the scale of the University of North Texas the system would need more processing power to run the backend with the number of simultaneous scans and data pushing/retrieval. The exact cost would depend on whether the university purchased their own hardware or used a third-party service provider.
- Staff
 - At minimum, we believe at least one developer and one system admin would be needed to properly maintain the system. Though, we would recommend having two developers, two system administrators, at least three support staff members, and a project manager if the system is operating on a large scale.

Risk & Contingency Plan

Potential Risk	Planned Mitigation Strategy
Facial Recognition Inaccuracies	DeepFace should be recalibrated each quarter using updated student images. Consistent classroom lighting and conditions should be maintained. Teachers will be able to manually modify attendance records if the recognition process fails.
Render Backend Downtime	In the event that our backend service provider, Render, goes down a backup deployment should be considered using another service provider. Uptime monitoring and alert notifications should be configured on Render.
Data Breach	Data should be encrypted as it moves and while stored. Make sure Firebase security rules are in place and use HTTPS only communication. Monthly security audits should be conducted.
Firebase Downtime/Data Corruption	Daily backups should be made. Verified backups should be stored offsite and data integrity checks should be made before restoring.
Poor System Performance	Quarterly performance testing should be conducted. Attempts to optimize endpoints and components should be made accordingly. Monitor responses times on Render.
System Turnover	System documentation should be thorough and up to date. Everything should be stored in a shared repository and regular code reviews should be conducted.
Environmental Issues	Classroom conditions should be tested on a quarterly basis to ensure that proper lighting conditions. Teachers will be able to modify

	attendance records if any issues arise due to poor environmental conditions.
--	--